Irwin King
Jun Wang
Laiwan Chan
DeLiang Wang (Eds.)

LNCS 4233

Neural Information Processing

13th International Conference, ICONIP 2006
Hong Kong, China, October 2006
Proceedings, Part II

2 Part II

Springer

Lecture Notes in Computer Science 4233

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Irwin King Jun Wang Laiwan Chan
DeLiang Wang (Eds.)

Neural
Information Processing

13th International Conference, ICONIP 2006
Hong Kong, China, October 3-6, 2006
Proceedings, Part II

Springer

Volume Editors

Irwin King
Laiwan Chan
Chinese University of Hong Kong
Department of Computer Science and Engineering
Shatin, New Territories, Hong Kong
E-mail:{king,lwchan}@cse.cuhk.edu.hk

Jun Wang
Chinese University of Hong Kong
Department of Automation and Computer-Aided Engineering
Shatin, New Territories, Hong Kong
E-mail: jwang@acae.cuhk.edu.hk

DeLiang Wang
Ohio State University
Department of Computer Science and Engineering, Columbus, Ohio, USA
E-mail: dwang@cse.ohio-state.edu

Library of Congress Control Number: 2006933758

CR Subject Classification (1998): F.1, I.2, I.5, I.4, G.3, J.3, C.2.1, C.1.3, C.3

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-46481-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-46481-5 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11893257 06/3142 5 4 3 2 1 0

Preface

This book and its companion volumes constitute the Proceedings of the 13th International Conference on Neural Information Processing (ICONIP 2006) held in Hong Kong during October 3–6, 2006. ICONIP is the annual flagship conference of the Asia Pacific Neural Network Assembly (APNNA) with the past events held in Seoul (1994), Beijing (1995), Hong Kong (1996), Dunedin (1997), Kitakyushu (1998), Perth (1999), Taejon (2000), Shanghai (2001), Singapore (2002), Istanbul (2003), Calcutta (2004), and Taipei (2005). Over the years, ICONIP has matured into a well-established series of international conference on neural information processing and related fields in the Asia and Pacific regions. Following the tradition, ICONIP 2006 provided an academic forum for the participants to disseminate their new research findings and discuss emerging areas of research. It also created a stimulating environment for the participants to interact and exchange information on future challenges and opportunities of neural network research.

ICONIP 2006 received 1,175 submissions from about 2,000 authors in 42 countries and regions (Argentina, Australia, Austria, Bangladesh, Belgium, Brazil, Canada, China, Hong Kong, Macao, Taiwan, Colombia, Costa Rica, Croatia, Egypt, Finland, France, Germany, Greece, India, Iran, Ireland, Israel, Italy, Japan, South Korea, Malaysia, Mexico, New Zealand, Poland, Portugal, Qatar, Romania, Russian Federation, Singapore, South Africa, Spain, Sweden, Thailand, Turkey, UK, and USA) across six continents (Asia, Europe, North America, South America, Africa, and Oceania). Based on rigorous reviews by the Program Committee members and reviewers, 386 high-quality papers were selected for publication in the proceedings with the acceptance rate being less than 33%. The papers are organized in 22 cohesive sections covering all major topics of neural network research and development. In addition to the contributed papers, the ICONIP 2006 technical program included two plenary speeches by Shun-ichi Amari and Russell Eberhart. In addition, the ICONIP 2006 program included invited talks by the leaders of technical co-sponsors such as Wlodzislaw Duch (President of the European Neural Network Society), Vincenzo Piuri (President of the IEEE Computational Intelligence Society), and Shiro Usui (President of the Japanese Neural Network Society), DeLiang Wang (President of the International Neural Network Society), and Shoujue Wang (President of the China Neural Networks Council). In addition, ICONIP 2006 launched the APNNA Presidential Lecture Series with invited talks by past APNNA Presidents and the K.C. Wong Distinguished Lecture Series with invited talks by eminent Chinese scholars. Furthermore, the program also included six excellent tutorials, open to all conference delegates to attend, by Amir Atiya, Russell Eberhart, Mahesan Niranjan, Alex Smola, Koji Tsuda, and Xuegong Zhang. Besides the regular sessions, ICONIP 2006 also featured ten special sessions focusing on some emerging topics.

ICONIP 2006 would not have achieved its success without the generous contributions of many volunteers and organizations. ICONIP 2006 organizers would like to express sincere thanks to APNNA for the sponsorship, to the China Neural Networks Council, European Neural Network Society, IEEE Computational Intelligence Society, IEEE Hong Kong Section, International Neural Network Society, and Japanese Neural Network Society for their technical co-sponsorship, to the Chinese University of Hong Kong for its financial and logistic supports, and to the K.C. Wong Education Foundation of Hong Kong for its financial support. The organizers would also like to thank the members of the Advisory Committee for their guidance, the members of the International Program Committee and additional reviewers for reviewing the papers, and members of the Publications Committee for checking the accepted papers in a short period of time. Particularly, the organizers would like to thank the proceedings publisher, Springer, for publishing the proceedings in the prestigious series of *Lecture Notes in Computer Science*. Special mention must be made of a group of dedicated students and associates, Haixuan Yang, Zhenjiang Lin, Zenglin Xu, Xiang Peng, Po Shan Cheng, and Terence Wong, who worked tirelessly and relentlessly behind the scene to make the mission possible. There are still many more colleagues, associates, friends, and supporters who helped us in immeasurable ways; we express our sincere thanks to them all. Last but not the least, the organizers would like to thank all the speakers and authors for their active participation at ICONIP 2006, which made it a great success.

October 2006

Irwin King
Jun Wang
Laiwan Chan
DeLiang Wang

Organization

Organizer

The Chinese University of Hong Kong

Sponsor

Asia Pacific Neural Network Assembly

Financial Co-sponsor

K.C. Wong Education Foundation of Hong Kong

Technical Co-sponsors

IEEE Computational Intelligence Society
International Neural Network Society
European Neural Network Society
Japanese Neural Network Society
China Neural Networks Council
IEEE Hong Kong Section

Honorary Chair and Co-chair

Lei Xu, Hong Kong
Shun-ichi Amari, Japan

Advisory Board

Walter J. Freeman, USA
Toshio Fukuda, Japan
Kunihiko Fukushima, Japan
Tom Gedeon, Australia
Zhen-ya He, China
Nik Kasabov, New Zealand
Okyay Kaynak, Turkey
Anthony Kuh, USA
Sun-Yuan Kung, USA
Soo-Young Lee, Korea
Chin-Teng Lin, Taiwan
Erkki Oja, Finland
Nikhil R. Pal, India
Marios M. Polycarpou, USA
Shiro Usui, Japan
Benjamin W. Wah, USA
Lipo Wang, Singapore
Shoujue Wang, China
Paul J. Werbos, USA
You-Shou Wu, China
Donald C. Wunsch II, USA
Xin Yao, UK
Yixin Zhong, China
Jacek M. Zurada, USA

General Chair and Co-chair

Jun Wang, Hong Kong

Laiwan Chan, Hong Kong

Organizing Chair

Man-Wai Mak, Hong Kong

Finance and Registration Chair

Kai-Pui Lam, Hong Kong

Workshops and Tutorials Chair

James Kwok, Hong Kong

Publications and Special Sessions Chair and Co-chair

Frank H. Leung, Hong Kong

Jianwei Zhang, Germany

Publicity Chair and Co-chairs

Jeffrey Xu Yu, Hong Kong

Chris C. Yang, Hong Kong

Derong Liu, USA

Wlodzislaw Duch, Poland

Local Arrangements Chair and Co-chair

Andrew Chi-Sing Leung, Hong Kong

Eric Yu, Hong Kong

Secretary

Haixuan Yang, Hong Kong

Program Chair and Co-chair

Irwin King, Hong Kong

DeLiang Wang, USA

Program Committee

Shigeo Abe, Japan
Peter Andras, UK
Sabri Arik, Turkey
Abdesselam Bouzerdoum, Australia
Ke Chen, UK
Liang Chen, Canada
Luonan Chen, Japan
Zheru Chi, Hong Kong
Sung-Bae Cho, Korea
Sungzoon Cho, Korea
Seungjin Choi, Korea
Andrzej Cichocki, Japan
Chuangyin Dang, Hong Kong
Wai-Keung Fung, Canada
Takeshi Furuhashi, Japan
Artur dAvila Garcez, UK
Daniel W.C. Ho, Hong Kong
Edward Ho, Hong Kong
Sanqing Hu, USA
Guang-Bin Huang, Singapore
Kaizhu Huang, China
Malik Magdon Ismail, USA
Takashi Kanamaru, Japan
James Kwok, Hong Kong
James Lam, Hong Kong
Kai-Pui Lam, Hong Kong
Doheon Lee, Korea
Minho Lee, Korea
Andrew Leung, Hong Kong
Frank Leung, Hong Kong
Yangmin Li, Macau
Xun Liang, China
Yanchun Liang, China
Xiaofeng Liao, China
Chih-Jen Lin, Taiwan
Xiuwen Liu, USA
Bao-Liang Lu, China
Wenlian Lu, China
Jinwen Ma, China
Man-Wai Mak, Hong Kong
Sushmita Mitra, India
Paul Pang, New Zealand
Jagath C. Rajapakse, Singapore
Bertram Shi, Hong Kong
Daming Shi, Singapore
Michael Small, Hong Kong
Michael Stiber, USA
Ponnuthurai N. Suganthan, Singapore
Fuchun Sun, China
Ron Sun, USA
Johan A.K. Suykens, Belgium
Norikazu Takahashi, Japan
Michel Verleysen, Belgium
Si Wu, UK
Chris Yang, Hong Kong
Hujun Yin, UK
Eric Yu, Hong Kong
Jeffrey Yu, Hong Kong
Gerson Zaverucha, Brazil
Byoung-Tak Zhang, Korea
Liqing Zhang, China

Reviewers

Shotaro Akaho
Toshio Akimitsu
Damminda Alahakoon
Aimee Betker
Charles Brown
Gavin Brown
Jianting Cao
Jinde Cao
Hyi-Taek Ceong
Pat Chan
Samuel Chan
Aiyou Chen
Hongjun Chen
Lihui Chen
Shu-Heng Chen
Xue-Wen Chen
Chong-Ho Choi
Jin-Young Choi
M.H. Chu
Sven Crone
Bruce Curry
Rohit Dhawan
Deniz Erdogmus
Ken Ferens
Robert Fildes
Tetsuo Furukawa
John Q. Gan

Kosuke Hamaguchi
Yangbo He
Steven Hoi
Pingkui Hou
Zeng-Guang Hou
Justin Huang
Ya-Chi Huang
Kunhuang Huarng
Arthur Hsu
Kazushi Ikeda
Masumi Ishikawa
Jaeseung Jeong
Liu Ju
Christian Jutten
Mahmoud Kaboudan
Sotaro Kawata
Dae-Won Kim
Dong-Hwa Kim
Cleve Ku
Shuichi Kurogi
Cherry Lam
Stanley Lam
Toby Lam
Hyoung-Joo Lee
Raymond Lee
Yuh-Jye Lee
Chi-Hong Leung
Bresley Lim
Heui-Seok Lim
Hsuan-Tien Lin
Wei Lin
Wilfred Lin
Rujie Liu
Xiuxin Liu
Xiwei Liu
Zhi-Yong Liu
Hongtao Lu
Xuerong Mao
Naoki Masuda
Yicong Meng
Zhiqing Meng
Yutaka Nakamura
Nicolas Navet
Raymond Ng
Rock Ng
Edith Ngai
Minh-Nhut Nguyen
Kyosuke Nishida
Yugang Niu
YewSoon Ong
Neyir Ozcan
Keeneth Pao
Ju H. Park
Mario Pavone
Renzo Perfetti
Dinh-Tuan Pham
Tu-Minh Phuong
Libin Rong
Akihiro Sato
Xizhong Shen
Jinhua Sheng
Qiang Sheng
Xizhi Shi
Noritaka Shigei
Hyunjung Shin
Vimal Singh
Vladimir Spinko
Robert Stahlbock
Hiromichi Suetant
Jun Sun
Yanfeng Sun
Takashi Takenouchi
Yin Tang
Thomas Trappenberg
Chueh-Yung Tsao
Satoki Uchiyama
Feng Wan
Dan Wang
Rubin Wang
Ruiqi Wang
Yong Wang
Hua Wen
Michael K.Y. Wong
Chunguo Wu
Guoding Wu
Qingxiang Wu
Wei Wu
Cheng Xiang
Botong Xu
Xu Xu
Lin Yan
Shaoze Yan
Simon X. Yang
Michael Yiu
Junichiro Yoshimoto
Enzhe Yu
Fenghua Yuan
Huaguang Zhang
Jianyu Zhang
Kun Zhang
Liqing Zhang
Peter G. Zhang
Ya Zhang
Ding-Xuan Zhou
Jian Zhou
Jin Zhou
Jianke Zhu

Table of Contents – Part II

Pattern Classification

Distance Function Learning in Error-Correcting Output Coding Framework 1
Dijun Luo, Rong Xiong

Combining Pairwise Coupling Classifiers Using Individual Logistic Regressions 11
Nobuhiko Yamaguchi

The Novelty Detection Approach for Different Degrees of Class Imbalance 21
Hyoung-joo Lee, Sungzoon Cho

A Novel Multistage Classification Strategy for Handwriting Chinese Character Recognition Using Local Linear Discriminant Analysis 31
Lei Xu, Baihua Xiao, Chunheng Wang, Ruwei Dai

Prototype Based Classification Using Information Theoretic Learning ... 40
Th. Villmann, B. Hammer, F.-M. Schleif, T. Geweniger, T. Fischer, M. Cottrell

A Modal Symbolic Classifier for Interval Data 50
Fabio C.D. Silva, Francisco de A.T. de Carvalho, Renata M.C.R. de Souza, Joyce Q. Silva

Hough Transform Neural Network for Seismic Pattern Detection 60
Kou-Yuan Huang, Jiun-De You, Kai-Ju Chen, Hung-Lin Lai, An-Jin Don

Autonomous and Deterministic Clustering for Evidence-Theoretic Classifier 70
Chen Li Poh, Loo Chu Kiong, M.V.C. Rao

Bark Classification Based on Gabor Filter Features Using RBPNN Neural Network 80
Zhi-Kai Huang, De-Shuang Huang, Ji-Xiang Du, Zhong-Hua Quan, Shen-Bo Guo

A Hybrid Handwritten Chinese Address Recognition Approach 88
Kaizhu Huang, Jun Sun, Yoshinobu Hotta, Katsuhito Fujimoto, Satoshi Naoi, Chong Long, Li Zhuang, Xiaoyan Zhu

A Morphological Neural Network Approach for Vehicle Detection from High Resolution Satellite Imagery 99
Hong Zheng, Li Pan, Li Li

Secure Personnel Authentication Based on Multi-modal Biometrics Under Ubiquitous Environments 107
Dae-Jong Lee, Man-Jun Kwon, Myung-Geun Chun

Pattern Classification Using a Set of Compact Hyperspheres 116
Amir Atiya, Sherif Hashem, Hatem Fayed

Direct Estimation of Fault Tolerance of Feedforward Neural Networks in Pattern Recognition 124
Huilan Jiang, Tangsheng Liu, Mengbin Wang

A Fully Automated Pattern Classification Method of Combining Self-Organizing Map with Generalization Regression Neural Network ... 132
Chao-feng Li, Jun-ben Zhang, Zheng-you Wang, Shi-tong Wang

Comparison of One-Class SVM and Two-Class SVM for Fold Recognition 140
Alexander Senf, Xue-wen Chen, Anne Zhang

Efficient Domain Action Classification Using Neural Networks 150
Hyunjung Lee, Harksoo Kim, Jungyun Seo

A New Hierarchical Decision Structure Using Wavelet Packet and SVM for Brazilian Phonemes Recognition 159
Adriano de A. Bresolin, Adrião Duarte D. Neto, Pablo Javier Alsina

Face Analysis and Processing

A Passport Recognition and Face Verification Using Enhanced Fuzzy Neural Network and PCA Algorithm 167
Kwang-Baek Kim, Sungshin Kim

A Weighted FMM Neural Network and Its Application to Face Detection 177
Ho-Joon Kim, Juho Lee, Hyun-Seung Yang

Fast Learning for Statistical Face Detection 187
Zhi-Gang Fan, Bao-Liang Lu

Extraction of Discriminative Manifold for Face Recognition 197
Yanmin Niu, Xuchu Wang

Gender Classification Using a New Pyramidal Neural Network 207
S.L. Phung, A. Bouzerdoum

A Novel Model for Gabor-Based Independent Radial Basis Function Neural Networks and Its Application to Face Recognition 217
GaoYun An, QiuQi Ruan

Generalized PCA Face Recognition by Image Correction and Bit Feature Fusion 227
Huiyuan Wang, Yan Leng, Zengfeng Wang, Xiaojuan Wu

E-2DLDA: A New Matrix-Based Image Representation Method for Face Recognition 236
Fei Long, Huailin Dong, Ling Fan, Haishan Chen

Adaptive Color Space Switching Based Approach for Face Tracking 244
Chuan-Yu Chang, Yung-Chin Tu, Hong-Hao Chang

A New Subspace Analysis Approach Based on Laplacianfaces 253
Yan Wu, Ren-Min Gu

Rotation Invariant Face Detection Using Convolutional Neural Networks 260
Fok Hing Chi Tivive, Abdesselam Bouzerdoum

Face Tracking Algorithm Based on Mean Shift and Ellipse Fitting...... 270
Jianpo Gao, Zhenyang Wu, Yujian Wang

Improving the Generalization of Fisherface by Training Class Selection Using SOM^2 278
Jiayan Jiang, Liming Zhang, Tetsuo Furukawa

Image Processing

Image Registration with Regularized Neural Network 286
Anbang Xu, Ping Guo

A Statistical Approach for Learning Invariants: Application to Image Color Correction and Learning Invariants to Illumination 294
B. Bascle, O. Bernier, V. Lemaire

Limited Recurrent Neural Network for Superresolution Image Reconstruction 304
Yan Zhang, Qing Xu, Tao Wang, Lei Sun

Remote Sensing Image Fusion Based on Adaptive RBF Neural Network 314
Yun Wen Chen, Bo Yu Li

Active Contour with Neural Networks-Based Information Fusion Kernel 324
Xiongcai Cai, Arcot Sowmya

A Novel Split-and-Merge Technique for Error-Bounded Polygonal Approximation 334
Bin Wang, Chaojian Shi

Fast and Adaptive Low-Pass Whitening Filters for Natural Images 343
Ling-Zhi Liao, Si-Wei Luo, Mei Tian, Lian-Wei Zhao

An Exhaustive Employment of Neural Networks to Search the Better Configuration of Magnetic Signals in ITER Machine 353
Matteo Cacciola, Antonino Greco, Francesco Carlo Morabito, Mario Versaci

Ultra-Fast fMRI Imaging with High-Fidelity Activation Map 361
Neelam Sinha, Manojkumar Saranathan, A.G. Ramakrishnan, Juan Zhou, Jagath C. Rajapakse

A Fast Directed Tree Based Neighborhood Clustering Algorithm for Image Segmentation 369
Jundi Ding, SongCan Chen, RuNing Ma, Bo Wang

An Efficient Unsupervised Mixture Model for Image Segmentation 379
Pan Lin, XiaoJian Zheng, Gang Yu, ZuMao Weng, Sheng Zhen Cai

Speckle Reduction of Polarimetric SAR Images Based on Neural ICA ... 387
Jian Ji, Zheng Tian

Robust ICA Neural Network and Application on Synthetic Aperture Radar (SAR) Image Analysis 394
Jian Ji, Zheng Tian

Kernel Uncorrelated Discriminant Analysis for Radar Target Recognition 404
Ling Wang, Liefeng Bo, Licheng Jiao

SuperResolution Image Reconstruction Using a Hybrid Bayesian Approach 412
Tao Wang, Yan Zhang, Yong Sheng Zhang

Retrieval-Aware Image Compression, Its Format and Viewer Based Upon Learned Bases 420
Naoto Katsumata, Yasuo Matsuyama, Takeshi Chikagawa, Fuminori Ohashi, Fumiaki Horiike, Shun'ichi Honma, Tomohiro Nakamura

A Suitable Neural Network to Detect Textile Defects 430
Md. Atiqul Islam, Shamim Akhter, Tamnun E. Mursalin, M. Ashraful Amin

MPEG Video Traffic Modeling and Classification Using Fuzzy C-Means Algorithm with Divergence-Based Kernel 439
Chung Nguyen Tran, Dong-Chul Park

A Novel Sports Video Logo Detector Based on Motion Analysis 448
Hongliang Bai, Wei Hu, Tao Wang, Xiaofeng Tong, Changping Liu, Yimin Zhang

A Fast Selection Algorithm for Multiple Reference Frames in H.264/AVC 458
Qing-lei Meng, Chun-lian Yao, Bo Li

An Automotive Detector Using Biologically Motivated Selective Attention Model for a Blind Spot Monitor 466
Jaekyoung Moon, Jiyoung Yeo, Sungmoon Jeong, PalJoo Yoon, Minho Lee

Wavelet Energy Signature: Comparison and Analysis 474
Xiaobin Li, Zheng Tian

Image Fusion Based on PCA and Undecimated Discrete Wavelet Transform 481
Wei Liu, Jie Huang, Yongjun Zhao

Signal Processing

Speech Recognition with Multi-modal Features Based on Neural Networks 489
Myung Won Kim, Joung Woo Ryu, Eun Ju Kim

Speech Feature Extraction Based on Wavelet Modulation Scale for Robust Speech Recognition 499
Xin Ma, Weidong Zhou, Fang Ju, Qi Jiang

Fuzzy Controllers Based QoS Routing Algorithm with a Multiclass Scheme for Ad Hoc Networks 506
Chao Gui, Baolin Sun

Direction of Arrival Estimation Based on Minor Component Analysis Approach 515
Donghai Li, Shihai Gao, Feng Wang, Fankun Meng

Two-Stage Temporally Correlated Source Extraction Algorithm with Its Application in Extraction of Event-Related Potentials 523
Zhi-Lin Zhang, Liqing Zhang, Xiu-Ling Wu, Jie Li, Qibin Zhao

Bispectrum Quantification Analysis of EEG and Artificial Neural Network May Classify Ischemic States 533
Liyu Huang, Weirong Wang, Sekou Singare

An Adaptive Beamforming by a Generalized Unstructured Neural Network 543
Askin Demirkol, Levent Acar, Robert S. Woodley

Application of Improved Kohonen SOFM Neural Network to Radar Signal Sorting 553
Chuang Zhao, Yongjun Zhao

Unscented Kalman Filter-Trained MRAN Equalizer for Nonlinear Channels 560
Ye Zhang, Jianhua Wu, Guojin Wan, Yiqiang Wu

A Jumping Genes Paradigm with Fuzzy Rules for Optimizing Digital IIR Filters 568
Sai-Ho Yeung, Kim-Fung Man

Practical Denoising of MEG Data Using Wavelet Transform 578
Abhisek Ukil

Signal Restoration and Parameters' Estimation of Ionic Single-Channel Based on HMM-SR Algorithm 586
X.Y. Qiao, G. Li, L. Lin

Signal Sorting Based on SVC & K-Means Clustering in ESM Systems . . . 596
Qiang Guo, Wanhai Chen, Xingzhou Zhang, Zheng Li, Di Guan

Computer Vision

Camera Pose Estimation by an Artificial Neural Network 604
Ryan G. Benton, Chee-hung Henry Chu

Depth Perception of the Surfaces in Occluded Scenic Images 612
Baoquan Song, Zhengzhi Wang, Xin Zhang

Incremental Learning Method for Unified Camera Calibration 622
Jianbo Su, Wendong Peng

Implicit Camera Calibration by Using Resilient Neural Networks 632
Pinar Çivicioğlu, Erkan Beşdok

Implicit Camera Calibration Using an Artificial Neural Network 641
Dong-Min Woo, Dong-Chul Park

3D Freeform Surfaces from Planar Sketches Using Neural Networks 651
Usman Khan, Abdelaziz Terchi, Sungwoo Lim, David Wright, Sheng-Feng Qin

General Adaptive Transfer Functions Design for Volume Rendering by Using Neural Networks 661
Liansheng Wang, Xucan Chen, Sikun Li, Xun Cai

Real-Time Synthesis of 3D Animations by Learning Parametric Gaussians Using Self-Organizing Mixture Networks 671
Yi Wang, Hujun Yin, Li-Zhu Zhou, Zhi-Qiang Liu

Data Pre-processing

An Excellent Feature Selection Model Using Gradient-Based and Point Injection Techniques 679
D. Huang, Tommy W.S. Chow

Driven Forward Features Selection: A Comparative Study on Neural Networks 693
Vincent Lemaire, Raphael Féraud

Non-negative Matrix Factorization Based Text Mining: Feature Extraction and Classification 703
P.C. Barman, Nadeem Iqbal, Soo-Young Lee

Adaptive Parameters Determination Method of Pulse Coupled Neural Network Based on Water Valley Area 713
Min Li, Wei Cai, Zheng Tan

The Forgetting Gradient Algorithm for Parameter and Intersample Estimation of Dual-Rate Systems 721
Hui-zhong Yang, Jun Tian, Feng Ding

Intelligent System for Feature Extraction of Oil Slick in SAR Images: Speckle Filter Analysis 729
Danilo L. de Souza, Adrião D.D. Neto, Wilson da Mata

Feature Selection for Neural Networks Through Binomial Regression ... 737
Gecynalda Soares S. Gomes, Teresa B. Ludermir

Automated Parameter Selection for Support Vector Machine Decision Tree 746
Gyunghyun Choi, Suk Joo Bae

Message-Passing for Inference and Optimization of Real Variables on Sparse Graphs 754
K.Y. Michael Wong, C.H. Yeung, David Saad

Analysis and Insights into the Variable Selection Problem 764
Amir F. Atiya

Dimensionality Reduction of Protein Mass Spectrometry Data Using Random Projection 776
Chen Change Loy, Weng Kin Lai, Chee Peng Lim

Fault Tolerant Training of Neural Networks for Learning Vector Quantization 786
Takashi Minohara

Clustering with a Semantic Criterion Based on Dimensionality Analysis 796
Wenye Li, Kin-Hong Lee, Kwong-Sak Leung

Improved Clustering and Anisotropic Gradient Descent Algorithm for Compact RBF Network 806
Delu Zeng, Shengli Xie, Zhiheng Zhou

Clustering Massive High Dimensional Data with Dynamic Feature Maps 814
Rasika Amarasiri, Damminda Alahakoon, Kate Smith-Miles

Zoomed Clusters 824
Jean-Louis Lassez, Tayfun Karadeniz, Srinivas Mukkamala

Forecasting and Prediction

Predicting Chaotic Time Series by Boosted Recurrent Neural Networks 831
Mohammad Assaad, Romuald Boné, Hubert Cardot

Uncertainty in Mineral Prospectivity Prediction 841
Pawalai Kraipeerapun, Chun Che Fung, Warick Brown, Kok Wai Wong, Tamás Gedeon

Thermal Deformation Prediction in Machine Tools by Using Neural Network 850
Chuan-Wei Chang, Yuan Kang, Yi-Wei Chen, Ming-Hui Chu, Yea-Ping Wang

Fuzzy Time Series Prediction Method Based on Fuzzy Recurrent Neural Network 860
Rafik Aliev, Bijan Fazlollahi, Rashad Aliev, Babek Guirimov

Research on a Novel Method Diagnosis and Maintenance for Key Produce Plant Based on MAS and NN 870
Weijin Jiang, Xiaohong Lin

Nonlinear Hydrological Time Series Forecasting Based on the Relevance Vector Regression 880
Fang Liu, Jian-Zhong Zhou, Fang-Peng Qiu, Jun-Jie Yang, Li Liu

A Distributed Computing Service for Neural Networks and Its Application to Flood Peak Forecasting 890
Jun Zhu, Chunbo Liu, Jianhua Gong, Daojun Wang, Tao Song

Automatic Inference of Cabinet Approval Ratings by Information-Theoretic Competitive Learning 897
Ryotaro Kamimura, Fumihiko Yoshida

Radial Basis Function Neural Networks to Foresee Aftershocks in Seismic Sequences Related to Large Earthquakes 909
Vincenzo Barrile, Matteo Cacciola, Sebastiano D'Amico, Antonino Greco, Francesco Carlo Morabito, Francesco Parrillo

Motion Vector Prediction Using Frequency Sensitive Competitive Learning 917
HyungJun Kim

Forecasting the Flow of Data Packets for Website Traffic Analysis – ASVR-Tuned ANFIS/NGARCH Approach 925
Bao Rong Chang, Shi-Huang Chen, Hsiu Fen Tsai

A Hybrid Model for Symbolic Interval Time Series Forecasting 934
André Luis S. Maia, Francisco de A.T. de Carvalho, Teresa B. Ludermir

Peak Ground Velocity Evaluation by Artificial Neural Network for West America Region 942
Ben-yu Liu, Liao-yuan Ye, Mei-ling Xiao, Sheng Miao

Forecasting Electricity Demand by Hybrid Machine Learning Model 952
Shu Fan, Chengxiong Mao, Jiadong Zhang, Luonan Chen

Short-Term Load Forecasting Using Multiscale BiLinear Recurrent Neural Network with an Adaptive Learning Algorithm 964
Chung Nguyen Tran, Dong-Chul Park, Hwan-Soo Choi

A New Approach to Load Forecasting: Using Semi-parametric Method and Neural Networks 974
Abhisek Ukil, Jaco Jordaan

Research of Least Square Support Vector Machine Based on Chaotic Time Series in Power Load Forecasting Model 984
Wei Sun, Chenguang Yang

Neurodynamic and Particle Swarm Optimization

Solving Extended Linear Programming Problems Using a Class of Recurrent Neural Networks 994
Xiaolin Hu, Jun Wang

A Recurrent Neural Network for Non-smooth Convex Programming Subject to Linear Equality and Bound Constraints 1004
Qingshan Liu, Jun Wang

Neural Networks for Optimization Problem with Nonlinear Constraints 1014
Min-jae Kang, Ho-chan Kim, Farrukh Aslam Khan, Wang-cheol Song, Sang-joon Lee

A Novel Chaotic Annealing Recurrent Neural Network for Multi-parameters Extremum Seeking Algorithm 1022
Yun-an Hu, Bin Zuo, Jing Li

Improved Transiently Chaotic Neural Network and Its Application to Optimization 1032
Yao-qun Xu, Ming Sun, Meng-shu Guo

Quantum-Behaved Particle Swarm Optimization for Integer Programming 1042
Jing Liu, Jun Sun, Wenbo Xu

Neural Network Training Using Stochastic PSO 1051
Xin Chen, Yangmin Li

Hybrid Training of Feed-Forward Neural Networks with Particle Swarm Optimization 1061
Marcio Carvalho, Teresa B. Ludermir

Clonal Selection Theory Based Artificial Immune System and Its Application 1071
Hongwei Dai, Yu Yang, Yanqiu Che, Zheng Tang

A Hybrid Algorithm to Infer Genetic Networks 1079
Cheng-Long Chuang, Chung-Ming Chen, Grace S. Shieh

An Intelligent PSO-Based Control Algorithm for Adaptive Compensation Polarization Mode Dispersion in Optical Fiber Communication Systems 1090
Xiaoguang Zhang, Lixia Xi, Gaoyan Duan, Li Yu, Zhongyuan Yu, Bojun Yang

Prediction of Construction Litigation Outcome Using a Split-Step PSO Algorithm 1101
Kwok-wing Chau

Solving Multiprocessor Real-Time System Scheduling with Enhanced Competitive Scheme 1108
Ruey-Maw Chen, Shih-Tang Lo, Yueh-Min Huang

A Distributed Hybrid Algorithm for Optimized Resource Allocation Problem 1118
Kyeongmo Park, Sungcheol Kim, Chuleui Hong

A Swarm Optimization Model for Energy Minimization Problem of Early Vision 1128
Wenhui Zhou, Lili Lin, Weikang Gu

PSO-Based Hyper-Parameters Selection for LS-SVM Classifiers 1138
X.C. Guo, Y.C. Liang, C.G. Wu, C.Y. Wang

Training RBF Neural Networks with PSO and Improved Subtractive Clustering Algorithms 1148
JunYing Chen, Zheng Qin

Training RBF Neural Network Via Quantum-Behaved Particle Swarm Optimization 1156
Jun Sun, Wenbo Xu, Jing Liu

Discrete Particle Swarm Optimization and EM Hybrid Approach for Naive Bayes Clustering 1164
Jing-Hua Guan, Da-You Liu, Si-Pei Liu

Extended Particle Swarm Optimiser with Adaptive Acceleration Coefficients and Its Application in Nonlinear Blind Source Separation . . 1174
Ying Gao, Zhaohui Li, Hui Zheng, Huailiang Liu

Application of a Hybrid Ant Colony Optimization for the Multilevel Thresholding in Image Processing 1183
Yun-Chia Liang, Angela Hsiang-Ling Chen, Chiuh-Cheng Chyu

Author Index 1193

Distance Function Learning in Error-Correcting Output Coding Framework

Dijun Luo and Rong Xiong

National Lab of Industrial Control Technology, Zhejiang University, China

Abstract. This paper presents a novel framework of error-correcting output coding (ECOC) addressing the problem of multi-class classification. By weighting the output space of each base classifier which is trained independently, the distance function of decoding is adapted so that the samples are more discriminative. A criterion generated over the Extended Pair Samples (EPS) is proposed to train the weights of output space. Some properties still hold in the new framework: any classifier, as well as distance function, is still applicable. We first conduct empirical studies on UCI datasets to verify the presented framework with four frequently used coding matrixes and then apply it in RoboCup domain to enhance the performance of agent control. Experimental results show that our supervised learned decoding scheme improves the accuracy of classification significantly and betters the ball control of agents in a soccer game after learning from experience.

1 Introduction

Many supervised machine learning tasks can be cast as the problem of assigning patterns to a finite set of classes, which is often referred to as multi-class classification. Examples include optical character recognition (OCR) system addresses the problem of determining the digit value of an image, text classification, speech recognition, medical analysis, and situation determination in robot control etc.. Some of the well known binary classification learning algorithms can be extended to handle multi-class problems [4, 16, 17]. Recently it becomes a general approach to combine a set of binary classifiers to solve a multi-class problem.

Dietterich and Bakiri [7] presented a typical framework of this approach, which is known as error-correcting output coding (ECOC), or output coding in short. The idea of ECOC enjoys a significant improvement in many empirical experiments [7, 8, 1, 18, 3, 2].

The methods of ECOC previously discussed, however, are based on a predefined output code and a fixed distance function. In this case, a predefined code is used to encode the base learners, and the predefined output code and a distance function is employed to compute the discriminative function, according to which a testing instance is assigned to some class. Crammer and Singer argued that the complexity of the induced binary problems would be ignored due to the predefinition of the output code. Hence a learning approach of designing an output code is presented [5].

This paper illustrates another way of adapting the decoding process of ECOC framework by learning approach which yields a significant improvement of multi-class classification in several empirical experiments. The major idea is redefining the distance

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1–10, 2006.

function by rescaling the output space of every base learner which is trained independently. By employing the idea of Vapnik's support vector machines (SVMs) we define a criteria as the sum of empirical hinge loss and the regularization with a trade-off factor between them. The criteria is generated over the *Extended Pair Samples (EPS)* which contain a subset of pair-instances as ranking SVMs.

Two experiments are conducted for validation of the performance of our method. The first is on UIC Repository and the second is on RoboCup domain. The experimental results show that our method outperforms the existing approaches significantly.

2 ECOC Framework

In ECOC framework, all base classifiers are trained independently. This training scheme ignores the dataset distribution and the performance of each base classifier. Though some probability based decoding methods are introduced in [14], the following problem remains unsolved: the criterion of a good is not well defined. Therefore, what is a better or best decoding function is not clear. In this paper, we illustrate a clear scheme of defining an optimal decoding function. The method proposed in this paper is different from finding an optimal decoding matrix which is first used by Crammer & Singer [5], and is probably much more efficient, because the optimization space is much simpler than that used in Crammer & Singer's method.

2.1 Scheme of Error-Correcting Output Coding

Let $S = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ denotes a set of training data where each instance x_i belongs to a domain $\mathcal{X}$ and each label y_i belongs to a set of labels representing categories $\mathcal{Y} = \{1, 2, ..., k\}$, and N is the number of instances. A multi-class classifier $H : \mathcal{X} \mapsto \mathcal{Y}$ is a function that maps an instance x in $\mathcal{X}$ into a label y in $\mathcal{Y}$.

A typical ECOC method is conducted as follows,

(1) **Encoding:** A codeword M is defined. M is a matrix of $k \times n$ size over $\{-1, 0, +1\}$ where k is size of label $\mathcal{Y}$ set and n is number of binary classifiers. Each row of M correspond to a category and each column corresponds to a binary classifier. The n binary base classifiers are denoted as $h_1(x), h_2(x), ..., h_n(x)$.

Several families of codes have been proposed and tested so far for encoding, such as, comparing each category against the rest , comparing all pairs of categories (one-against-one), employing the random code, and employing the Hadamard code [7, 9, 11].

(2) **Base classifier construction:** A dichotomy of samples is created for each base classifier. The dichotomies vary according to classifiers. If $M_{y,s} = -1$, we take all the instances labeled y as negative samples in training set of the base learner h_s . If $M_{y,s} = 1$, we take all the instances labeled y as positive samples in training set of the base learner h_s. If $M_{y,s} = 0$, the instances are ignored. SVMs can be used as the model of base classifier.

(3) **Decoding:** Given an instance x, a vector of binary labels is generated from all the base classifiers $\mathcal{H}(x) = (h_1(x), h_2(x), ..., h_n(x))$. We then compare the vector with each row of the matrix M (each category). A final classification decision is made using the discriminate function as follows,

$$H(x) = \arg\min_{y\in\mathcal{Y}} \mathcal{F}(x,y) \tag{1}$$

$$\mathcal{F}(x,y) = D(M_y, \mathcal{H}(x)) \tag{2}$$

where $D(u,v)$ is distance function between vectors u and v, and M_y is the row y of the code matrix M. Consequently, the label of x is predicted to be y if the output of base classifiers is the 'closest' to the row of M_y.

2.2 ECOC Framework with Decoding Learning

A lot of empirical experiments show that ECOC enjoys a significant improvement [7, 8, 1, 18, 3, 2]. One, however, argues that ECOC suffers the following problem [15]: Hamming decoding scheme ignores the confidence of each classifier in ECOC and this confidence is merely a relative quantity which means using a linear loss base distance function in decoding may introduce some bias in the final classification in the sense that classifiers with a larger output range will receive a higher weight. Thus both Hamming distance function and simple loss base distance function have disadvantage. In [15] a probability based decoding distance function is proposed. The relation between an optimal criterion and the parameters of the distance function is not well defined. Therefore in fact, the introduction of probability based distance function is just an approximation of an optimal decoding. This paper presents a learning approach to searching an optimal distance function for ECOC decoding which will overcome the problem suffered by previous work.

3 Distance Function Learning

In this paper we present a novel algorithm of multi-class classification (which is termed OC.MM) by introducing the max margin distance function learning in ECOC.

We rewrite the distance function as the following form,

$$D(u,v) = \sum_{s=1}^{n} d(u_s, v_s).$$

which implies that the distance or similarity is composed of each dimension independently. This property holds in most of the existing distance function include hamming distance and linear distance. In our distance learning approach, we assign each dimension of the output of base learner a weight, so that the output space of $\mathcal{H}(x)$ is rescaled. The larger the distance is, the less the similarity is. Thus we can equivalently consider a weighted version of similarity function as,

$$K(u,v) = \sum_{s=1}^{n} w_s k(u_s, v_s).$$

Consequently the final classification hypothesis is

$$y = H(x) = \arg\max_{y} \left(\sum_{s=1}^{n} w_s k(M_{y,s}, h_i(x)) \right) \tag{3}$$

We denote

$$F(x, y; w) = \sum_{s=1}^{n} w_s k(M_{y,s}, h_i(x)) = \langle w, \sigma_y(x) \rangle, \tag{4}$$

where $w = [w_1, w_2, ...w_n]$,$\sigma_y = [k(M_{y,1}, h_1(x)), k(M_{y,2}, h_2(x)), ...k(M_{y,n}, h_n(x))]$, and $\langle u, v\rangle$ denotes the inner product of u and v.

In order to illustrate our method of max margin decoding distance function, we first define the Extended Pair Samples (EPS) as follows,

$$S^{EPS} = \left\{ \left([\sigma_{y_k}(x_i), \sigma_{y_j}(x_i)], z_{i,y_k,y_j} = \begin{cases} 1, y_k = y_i, y_j \neq y_i \\ -1, y_j = y_i, y_k \neq y_i \end{cases} \right) : (x_i, y_i) \in S \right\}. \tag{5}$$

3.1 Primal QP Problem and Dual Problem

We consider the multi-class classification problem as a ranking one. An instance is correctly classified if a pattern $\sigma_{y_i}(x_i)$ ranks first in a subset of S^{EPS} given any instance x_i. That is

$$F(x_i, y_i; w) \geq F(x_i, y; w), \forall y \in \mathcal{Y}, y \neq y_i. \tag{6}$$

Then the criteria of OC.MM is as follows,

$$\min_{w} \sum_{\omega \in S^{EPS}} \left[1 - \langle w, \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i)\rangle\right]_+ + \lambda \|w\|^2 \tag{7}$$

where $\omega = \left([\sigma_{y_k}(x_i), \sigma_{y_j}(x_i)], z_{i,y_k,y_j}\right)$, $[z]_+ = \max(0, z)$ and λ is a wight between the regularization and the hinge loss. Instead of solving the above optimization, we solve the following equivalent one [10],

$$\frac{1}{2} \min_{w} + C \sum_{\omega \in S^{EPS}} \xi_\omega \tag{8}$$

$$s.t.$$

$$z_{i,y_j,y_k} \langle w, \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i)\rangle \geq 1 - \xi_\omega, \xi_\omega \geq 0.$$

Employing the Lagrangian multiplier method, the Lagrange function of (8) can be written as,

$$\mathcal{L}(w, \alpha, \xi, \zeta) = \frac{1}{2} \min_{w} + C \sum_{\omega \in S^{EPS}} \xi_\omega - \sum_{\omega \in S^{EPS}} \zeta_\omega \xi_\omega$$

$$- \sum_{\omega \in S^{EPS}} \alpha_\omega \left(z_{i,y_j,y_k} \langle w, \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i)\rangle + 1 - \xi_\omega \right). \tag{9}$$

According to KKT conditions,

$$\frac{\partial \mathcal{L}_D}{\partial w_s} = 0 \Longleftrightarrow w_s = \sum_{\omega \in S^{EPS}} \alpha_\omega z_{i,y_j,y_k} \left(\sigma_{y_j}(x_i) - \sigma_{y_k}(x_i) \right) \tag{10}$$

$$\frac{\partial \mathcal{L}_D}{\partial \xi_\omega} = 0 \Longleftrightarrow C - \alpha_\omega - \zeta_\omega = 0 \tag{11}$$

Since $\zeta_\omega > 0$, optimization problem (8) reduces to a box constraint $0 \leq \alpha_\omega \leq C$. By substituting (10) and (11) into (9), we obtain the Lagrangian dual objective (12),

$$\mathcal{L}_D(\alpha) = \sum_{\omega \in S^{EPS}} \alpha_\omega -$$

$$\frac{1}{2} \sum_{\omega \in S^{EPS}} \sum_{\omega' \in S^{EPS}} \alpha_\omega \alpha_{\omega'} z_{i,y_j,y_k} z_{i',y'_k,y'_j} \langle \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_{y'_j}(x_{i'}) - \sigma_{y'_k}(x_{i'}) \rangle, \tag{12}$$

where $\omega = \left([\sigma_{y_k}(x_i), \sigma_{y_j}(x_i)], z_{i,y_k,y_j}\right)$ and $\omega' = \left([\sigma_{y'_k}(x_{i'}), \sigma_{y'_j}(x_{i'})], z_{i',y'_k,y'_j}\right)$.

The solution of the dual QP is thus characterized by

$$\max_\alpha \mathcal{L}_D(\alpha)$$

$$s.t.$$

$$0 \leq \alpha_\omega \leq C, \forall \omega = \left([\sigma_{y_k}(x_i), \sigma_{y_j}(x_i)], z_{i,y_k,y_j}\right) \in S^{EPS} \tag{13}$$

We notice that it is easy to generalize the linear learning algorithm to non-linear cases using kernel functions. Substituting (10) into (4), the following is derived,

$$F(x, y, w) = \sum_{\omega \in S^{EPS}} \alpha_\omega z_{i,y_j,y_k} \langle \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_y(x) \rangle. \tag{14}$$

Replacing the inner products $\langle \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_y(x) \rangle$ and $\langle \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_{y'_j}(x_{i'}) - \sigma_{y'_k}(x_{i'}) \rangle$ with $K\left(\sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_y(x)\right)$ and $K\left(\sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_{y'_j}(x_{i'}) - \sigma_{y'_k}(x_{i'})\right)$, where $K(u, v)$ is a kernel function, one can make the generalization. Then we obtain a nonlinear weighted decoding distance optimization criterion of algorithm OC.MM as follows,

$$\mathcal{L}_D(\alpha) = c^T \alpha - \alpha^T \Lambda \alpha \tag{15}$$

where Λ is the kernel matrix containing all the kernel values over S^{EPS} and $c = [1, 1, ...1]$. The final classification hypothesis as following,

$$y = \arg\max_y F(x, y, w) = \sum_{\omega \in S^{EPS}} \alpha_\omega z_{i,y_j,y_k} K\left(\sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_y(x)\right). \tag{16}$$

3.2 Effective Training Scheme

To faster the convergence of the algorithm above we introduce an effective training scheme which is shown in Algorithm 1.

The algorithm above is implemented by modifying Joachims' SVM^{light} [12].

Algorithm 1. Effective algorithm for solving OC.MM

```
Input:$S^{EPS}$, $C$, $\epsilon$, $p$
$S_i \leftarrow \Phi, i = 1, 2, ..., N$
Randomly choose instances from $S^{EPS}$ into $S_i$ with probability $p$.
repeat
    for all $i$ such that $0 \leq i \leq N$ do
        $Q(y) = 1 - \sum_{\omega \in S^{EPS}} \alpha_\omega z_{i,y_j,y_k} K\left(\sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_y(x)\right)$
        $\hat{y}_i = \arg\max_{y \in \mathcal{Y}} Q(y)$
        $\hat{Q} = Q(\hat{y})$
        $\xi_i = \left[\max_{y \in S_i} Q(y)\right]_+$
        if $\hat{Q} > \xi_i + \epsilon$ then
            $S_i \leftarrow S_i \cup \left([\sigma_{y_i}(x_i), \sigma_{\hat{y}}(x_i)], z_{i,y_i,\hat{y}}\right) \cup \left([\sigma_{\hat{y}}(x_i), \sigma_{y_i}(x_i)], z_{i,\hat{y},y_i}\right)$
            $\alpha_{S_w} \leftarrow$optimize dual over $S_w = \cup_i S_i$
        end if
    end for
until $S_w$ dose not change.
```

4 Evaluations

Two experiments are conducted to evaluate the performance of the approach of OC.MM proposed in this paper. The first is conducted on 10 datasets selected from the UCI Repository. The second test-bed from the study on the application of our method in the domain of agent control.

4.1 Experimental Result on UCI Repository

We choose 11 datasets on UCI Repository to conduct this experiment. The datasets statistics are given in Table 1.

Four frequently used coding matrixes are applied in the experiments: one vs one, one vs rest, Hadamard, and random. In each we run SVM^{light} [12] as the baseline. We set the random code to have 2k columns for the problem which has k classes. The entry in

Table 1. Statistics on UCI datasets

Problem	#train	#test	#Attribute	#class
Glass	214	0	9	6
Segment	2310	0	19	7
Pendigits	7494	3498	16	10
Yeast	1484	0	8	10
Vowel	528	0	10	11
Shuttle	43500	14500	9	7
Soybean	307	376	35	19
Wine	178	0	13	3
Dermatology	366	0	34	6
Vehicle	846	0	18	4

matrix is set to be -1 or +1 uniformly at random. Hadamard code is generated by the following scheme,

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, H_{n+1} = \begin{pmatrix} H_n & H_n \\ H_n & -H_n \end{pmatrix}.$$

For the base line we chose SVMs with the RBF kernels $K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}$ as the base classifiers. We tune the cost parameters C in set $C = [2^{-6}, 2^{-5}, ..., 2^8]$ and γ from set $\gamma = [2^{-10}, 2^{-9}, ..., 2^4]$, and choose the best result for each algorithm.For the datasets in which the number of training instances is less than 2000 or there are no testing data, we use a 10-fold cross validation.

EPS of our algorithm is generated from the same output of SVM^{light}. Thus the accuracy of SVM^{light} is that of OC.MM without learning and with equal weights. Experimental results are shown in Table 2 from which we can see a significant improvement after applying our algorithm. Out of the $11 \times 4 = 44$ results, OC.MM outperforms SVM^{light} in 35; they draw in the rest.

Table 2. Prediction accuracy of SVM^{light} (SVM) and OC.MM on UCI datasets

	One-vs-one		*One-vs-rest*		*Random*		*Hadamard*	
Problem	SVM	OC.MM	SVM	OC.MM	SVM	OC.MM	SVM	OC.MM
Satimage	0.9204	0.9204	0.8933	0.8979	0.9176	0.9191	0.9159	0.9182
Glass	0.6728	0.6962	0.6822	0.6962	0.7009	0.7009	0.6822	0.7056
segmentation	0.9718	0.9735	0.9528	0.9640	0.9606	0.9671	0.9606	0.9645
Pendigits	0.9958	0.9958	0.9940	0.9958	0.9952	0.9952	0.9950	0.9952
Yeast	0.5923	0.5923	0.4791	0.4791	0.5404	0.5606	0.4696	0.4716
Vowel	0.9886	0.9886	0.9772	0.9791	0.9753	0.9829	0.9753	0.9772
Shuttle	0.9970	0.9970	0.9969	0.9971	0.9971	0.9972	0.9971	0.9972
Soybean	0.9414	0.9428	0.9136	0.9341	0.9443	0.9487	0.9428	0.9502
Wine	0.9490	0.9490	0.9157	0.9550	0.9157	0.9550	0.9325	0.9438
Dermatology	0.9726	0.9754	0.9480	0.9644	0.9672	0.9726	0.9453	0.9754
Vehicle	0.8475	0.8475	0.8392	0.8534	0.8498	0.8747	0.8333	0.8546

4.2 Empirical Study on Agent Control

We conduct the second experiment on the task of opponent action prediction to evaluate the effectiveness of our algorithm. The test-bed is RoboCup robot soccer simulation which offers a special type of benchmark requiring real-time sensor evaluation and decision making, acting in highly dynamic and competitive environment etc. [13]. In this paper we focus on the task of predicting the action of an opponent possessing the ball in such an environment. This is an important subtask in RoboCup soccer game which enables our agents to model the opponents' action pattern. For example, when our agents are defending in front of our goal, it is more like to disorganize the opponent's attack if the agents could accurately predict who will the opponent possessing the ball will pass to. The prediction is viewed as a multi-class classification problem on the target space as follows,

$$A = \{pass\_to\_teammate\_1, ..., pass\_to\_teammate\_11, Dribble\}$$

The features of state includes

- The absulute position the ball in current cycle and immediately previous cycle.
- The relative position of all players with respect to the ball in current cycle and immediately previous cycle.

The positions of ball are presented in Cartesian coordinates and all relative positions are presented in Polar coordinates. Figure 1 illustrates an instance at the moment of an opponent player possessing the ball in a soccer game.

We extract training and testing data from 99 games played between our agents and the champion of RoboCup 2004. We conduct these experiments to enable our agents to learn from the experience of playing with an opponent team. The statistics of these experiments is shown in Table 3 and Figure 2.

In this experiment, we also use the parameters tuning scheme applied in the experiment conducted on UCI datasets above. The experimental results are illustrated in Figure 3. In all four coding matrixes, our method outperforms SVM^{light}.

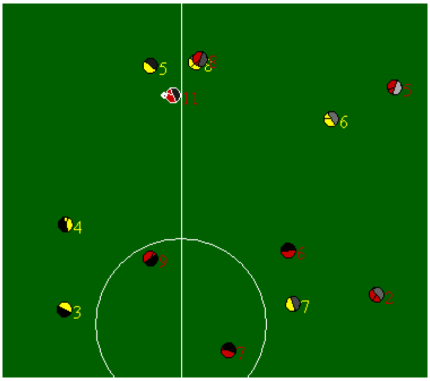


Fig. 1. Task of Robot Pass. Player 11 of opponent team (shown in red color) is possessing the ball, the task of RobotPass is to determine the next action of the player possessing ball. The potential action of opponent player 11 is dribbling or pass the ball to it's teammate 9 in the current situation.

Table 3. Statistics on RobotPass

	#games	#instance	#pass	#dribbling
Train	88	91109	16689	74420
Test	11	11440	2058	9382

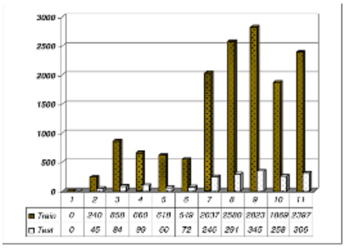


Fig. 2. The receive-passing frequency of each opponent in both training and testing data

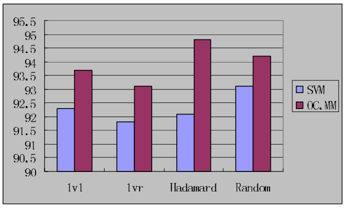


Fig. 3. Classification accuracy of SVM^{light} (SVM) and our method (OC.MM)

5 Conclusions and Future Works

In this paper we present a novel version of ECOC framework which significantly boosts the performance of multi-class classification. We give a criteria of ECOC decoding by defining a global loss based on the empirical loss and regularization over the Extended Pair Samples. Empirical results on both UCI datasets and the task of opponent action prediction in RoboCup domain show the utility of our algorithm. We also notice that the performance improvement is more significant on the datasets which have more classes. This might be due to the limitation of conventional ECOC framework on complex data while it is overcome in our approach.

In spite of the presented effective training scheme of OC.MM, a large scale quadratic programming problem is still time-consuming. Although the training can be conducted off-line, the efficiency of optimization remains to be further improved in order to make our algorithm more practical in very large datasets. Another direction of future work is to conduct further statistical analysis on the OC.MM algorithm. In this novel ECOC framework, the problem of codewords selection remains open. But the introduction of decoding margin provides a potential direction of further statistical analysis such as upper bound of generalization using statistical learning theorems.

Acknowledgement

This work is supported by the National Science Foundation of China (NSFC) No. 60305010 and No. 60421002.

References

[1] Aha, D. W. (1997). Cloud classification using error-correcting output codes. *Artificial Intelligence Applications: Natural Science, Agriculture, and Environmental Science*, *11*, 13–28.

[2] Allwein, E., Schapire, R., & Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *achine Learning: Proceedings of the Seventeenth International Conference. Artificial Intelligence Research, 2*, 263-286.

[3] Berger, A. (1999). Error-correcting output coding for text classification. In *IJCAI'99: Workshop on Machine Learning for Information Filtering*, In Berlin: Springer Verlag.

[4] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. Belmont,CA: Wadsworth & Brooks.

[5] Crammer, K.,Singer, Y. (2002). On the Learnability and Design of Output Codes for Multiclass Problems. *Machine Learning* 47(2-3):201-233.

[6] Crammer, K. & Singer. Y. (2001). On the algorithmic implementation of multiclass kernel-based machines. *Journal of Machine Learning Research, 2(Dec)*:265–292.

[7] Dietterich, T. G., & Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research, 2,* 263-286.

[8] Dietterich, T., & Kong, E. B. (1995). –Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. Technical report, Oregon State University. Available via the WWW at http://www.cs.orst.edu:80/ tgd/cv/tr.html.

[9] Hastie, T. & Tibshirani, R. (1998). Classification by pairwise coupling. In *Advances in Neural Information Processing Systems*, volume 10. MIT Press.

[10] Hastie, T., Tibshirani, R., & Friedman, J. (2001). The Elements of Statistical Learning: data mining, inference and prediction. Springer-Verlag.

[11] Hsu, C.-W. & Lin, C.-J. (2002). A comparison of methods for multi-class support vector machines , *IEEE Transactions on Neural Networks, 13,* 415-425.

[12] Joachims, T. (2002). Optimizing Search Engines Using Clickthrough Data. *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, ACM.

[13] Kuhlmann, G., & Stone, P. (2004). Progress in learning 3 vs. 2 keepaway. In Polani, D.; Browning, B.; Bonarini, A.; and Yoshida, K., eds., RoboCup-2003: Robot Soccer World Cup VII.

[14] Passerini, A., Pontil, M., & Frasconi, P.(2004). New results on error correcting output codes of kernel machines. *IEEE Transactions on Neural Networks, 15(1)*:45-54.

[15] Passerini, A.,Pontil, M., & Frasconi, F. (2002). From Margins to Probabilities in Multiclass Learning Problems. *ECAI:* 400-404.

[16] Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann.

[17] Rumelhart, D. E., Hinton, G. E.,&Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, & J. L. McClelland (Eds.), Parallel distributed processing-explorations in the microstructure of cognition (ch. 8, pp. 318-362). Cambridge, MA: MIT Press.

[18] Schapire, R. E. (1997). Using output codes to boost multiclass learning problems. In Machine Learning. *Proceedings of the Fourteenth International Conference* (pp. 313-321).

Combining Pairwise Coupling Classifiers Using Individual Logistic Regressions

Nobuhiko Yamaguchi

Faculty of Science and Engineering, Saga University, Saga-shi, 840–8502 Japan

Abstract. Pairwise coupling is a popular multi-class classification approach that prepares binary classifiers separating each pair of classes, and then combines the binary classifiers together. This paper proposes a pairwise coupling combination strategy using individual logistic regressions (ILR-PWC). We show analytically and experimentally that the ILR-PWC approach is more accurate than the individual logistic regressions.

1 Introduction

The object of this paper is to construct K-class classifiers. It is often easier to construct a multi-class classifier by combining multiple binary classifiers than directly construct a multi-class classifier. For example, AdaBoost [1] and support vector machines (SVM) algorithm [2] [3] are basically binary classifiers, and it is difficult to directly expand into multi-class classifiers. Typically, in such case, multi-class classifiers are constructed by decomposing the multi-class problem into multiple binary classification problems that can be handled by the AdaBoost and SVM algorithm. In addition, neural networks [4] are also binary classifiers since each output neuron separates a class from all other classes.

There are many ways to decompose a multi-class problem into multiple binary classification problems: one-per-class, individual logistic regressions [5] and pairwise coupling [6] [7]. One-per-class is one of the simplest approaches for decomposing the multi-class problem. The one-per-class approach prepares K binary classifiers, each of which separates a class from all other classes, and then constructs a multi-class classifier by combining the K binary classifiers. Next, the individual logistic regressions prepare $K-1$ binary classifiers, each of which separates a class i from an arbitrary selected baseline class j. Finally, the pairwise coupling approach prepares $K(K-1)/2$ binary classifiers, each of which separates a class i from a class j. In this paper, we focus on the pairwise coupling approach, and propose a pairwise coupling combination strategy using the individual logistic regressions. In particularly, we investigate the accuracy of our combination strategy in comparison with the individual logistic regressions.

Hastie and Tibshirani [7] show experimentally that the pairwise coupling approach is more accurate than the one-par-class approach. However the accuracy of the pairwise coupling approach has not been almost investigated theoretically. This is because that the combination strategy of the pairwise coupling approach is nonlinear and iterative. On the other hand, individual logistic regressions had

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 11–20, 2006.

the same combination problem, but Begg and Gray [5] proposed a simple linear and non-iterative combination strategy with consistent property. For these reasons, we propose a pairwise coupling combination strategy using individual logistic regressions (ILR-PWC), and investigate the accuracy of our combination strategy. As a result, we show that our strategy constructs more accurate multi-class classifiers in comparison with the individual logistic regressions.

This paper is organized as follows. Section 2 explains the pairwise coupling approach proposed by Hastie and Tibshirani [7]. Section 3 explains individual logistic regressions. In section 4, we propose an extension of the pairwise coupling approach, called ILR-PWC, and compare the accuracy of the individual logistic regressions and our approach. Section 5 describes the experimental results.

2 Pairwise Coupling

2.1 Pattern Classification

In K-class classification problems, the task is to assign an input $\boldsymbol{x}_0$ to one of K classes. To solve the problems, we first estimate the posterior probability $p_i^* = P(Y_0 = i|\boldsymbol{x}_0)$ that a given input $\boldsymbol{x}_0$ belongs to a particular class i, with a training set $d = \{(\boldsymbol{x}_n, y_n) \mid 1 \leq n \leq N\}$. We then select the class with the highest posterior probability:

$$y_0 = \arg \max_{1 \leq i \leq K} p_i^* . \tag{1}$$

In the rest of this section, we consider to estimate the posterior probability p_i^* with the training set d.

2.2 Constructing Binary Classifiers

The structure of pairwise coupling is illustrated in Fig. 1. Pairwise coupling is a multi-class classification approach that prepares $K(K-1)/2$ binary classifiers r_{ij}, $1 \leq i \leq K,\ 1 \leq j < i$, and then estimates the posterior probabilities p_i^* by combining the binary classifiers together. The binary classifiers r_{ij} are trained so as to estimate pairwise class probabilities $\mu_{ij}^* = P(Y_0 = i \mid Y_0 = i \text{ or } Y_0 = j,\ \boldsymbol{x}_0)$. The estimates r_{ij} of μ_{ij}^* are available by training with the ith and jth classes of the training set:

$$d_{ij} = \{(\boldsymbol{x}_n, y_n) \mid y_n = i \text{ or } y_n = j,\ 1 \leq n \leq N\} . \tag{2}$$

Then, using all r_{ij}, the goal is to estimate $p_i^* = P(Y_0 = i|\boldsymbol{x}_0)$, $i = 1, \cdots, K$.

2.3 Estimating Posterior Probabilities

Here, we describe a method for estimating the posterior probabilities p_i^*, proposed by Hastie and Tibshirani [7]. First note that the probabilities μ_{ij}^* can be rewritten as

$$\mu_{ij}^* = P(Y_0 = i \mid Y_0 = i \text{ or } Y_0 = j,\ \boldsymbol{x}_0) = p_i^*/(p_i^* + p_j^*). \tag{3}$$

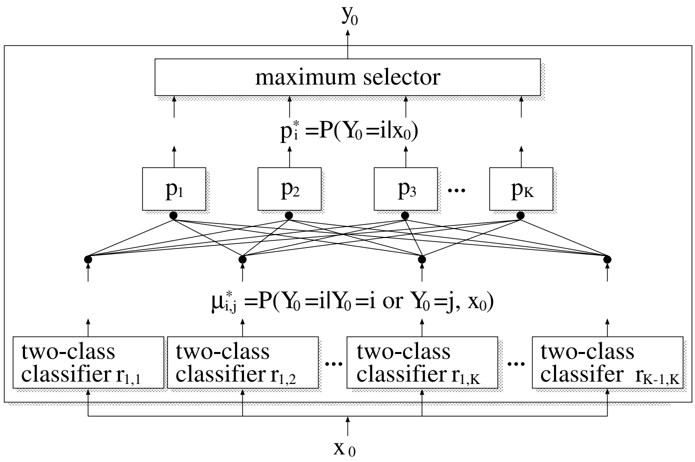

y0
maximum selector
p*i =P(Y0=i|x0)
p1
p2
p3
...
pK
μ*i,j =P(Y0=i|Y0=i or Y0=j, x0)
two-class classifier r1,1
two-class classifier r1,2
...
two-class classifier r1,K
...
two-class classifer rK-1,K
x 0

Fig. 1. Structure of pairwise coupling

```
Step 1. Initialize $p_i$ and compute coressponding $\mu_{ij}$.
Step 2. Repeat until conversence:
(a) For each $i = 1, \cdots, K$
        $p_i \leftarrow p_i \cdot \frac{\sum_{j \neq i}^{K} n_{ij} r_{ij}}{\sum_{j \neq i}^{K} n_{ij} \mu_{ij}}.$
(b) Renormalize the $p_i$.
(c) Recompute the $\mu_{ij}$.
```

Fig. 2. Algorithm for estimating posterior probabilities

From (3), they consider the model as follows:

$$\mu_{ij} = p_i / (p_i + p_j), \tag{4}$$

and propose to find the estimates p_i of p_i^* so that μ_{ij} are close to the observed r_{ij}. The closeness measure is the Kullback-Leibler (KL) divergence between r_{ij} and μ_{ij}:

$$l(p_1, \cdots, p_K) = \sum_{i=1}^{K} \sum_{j=i+1}^{K} n_{ij} \left[r_{ij} \log \frac{r_{ij}}{\mu_{ij}} + (1 - r_{ij}) \log \frac{1 - r_{ij}}{1 - \mu_{ij}} \right] \tag{5}$$

where n_{ij} is the number of elements in the training set d_{ij}. Hastie and Tibshirani [7] propose to find the estimates p_i that minimize the function l, and also propose to use an iterative algorithm to compute the p_i's as illustrated in Fig. 2.

3 Individual Logistic Regressions

The object of this paper is to propose a pairwise coupling combination strategy using individual logistic regressions [5]. The individual logistic regressions are

K-class classification approaches that combine $K-1$ binary classifiers. In this section, we describe the individual logistic regressions, and in the next section, we propose a pairwise coupling combination strategy using the individual logistic regressions (ILR-PWC).

3.1 Background

Multinomial logistic regressions [8] are popular approaches for solving multi-class classification problems. However, at the time when the individual logistic regressions were proposed, most statistical software packages included only simple binary logistic regressions, but did not include the multinomial logistic regressions. For this reason, Begg and Gray [5] proposed the individual logistic regressions which approximate the multinomial logistic regressions by combining multiple binary logistic regressions. They show that the approximation algorithm is not maximum likelihood but is consistent [5]D In addition, some experiments [5] [9] show that the efficiency loss of the approximation is small. For these reasons, the individual logistic regressions are still used to approximate the multinomial logistic regressions.

The rest of this section is organized as follows. Section 3.2 and 3.3 describe the logistic regressions and multinomial logistic regressions, respectively. In section 3.4, we describe a method for approximating the multinomial logistic regressions by using the individual logistic regressions.

3.2 Logistic Regressions

Logistic regressions are one of the most widely used techniques for solving binary classification problems. In the logistic regressions, the posterior probabilities p_i^*, $i \in \{1, 2\}$, are represented as the following:

$$\pi_1 = \frac{\exp(\eta)}{1+\exp(\eta)}, \quad \pi_2 = 1 - \pi_1 \tag{6}$$

where η is a function of an input $\boldsymbol{x}_0$. For example, η is a linear function of the input $\boldsymbol{x}_0$, that is,

$$\eta = \boldsymbol{\alpha}^T \boldsymbol{x}_0 + \beta, \tag{7}$$

and the parameters $\boldsymbol{\alpha}$, β are estimated by the maximum likelihood method. In this paper, η is an arbitrary function of $\boldsymbol{x}_0$. Note that if you choose an appropriate η, the model in (6) can represent some kinds of binary classification systems, such as neural networks, logitBoost [10], etc.

3.3 Multinomial Logistic Regressions

Multinomial logistic regressions are one of the techniques for solving multi-class classification problems. In the multinomial logistic regressions, the posterior probabilities p_i^*, $i \in \{1, \cdots, K\}$, are represented as the following:

$$\pi_i^j = \begin{cases} \dfrac{\exp(\eta_i^j)}{1 + \sum_{k \neq j}^{K} \exp(\eta_k^j)} & \text{if} \quad i \neq j \\[2ex] \dfrac{1}{1 + \sum_{k \neq j}^{K} \exp(\eta_k^j)} & \text{otherwise} \end{cases} \tag{8}$$

where j is a baseline class and η_i^j is a function of an input $\boldsymbol{x}_0$. For example, η_i^j is a linear function of the input $\boldsymbol{x}_0$, that is,

$$\eta_i^j = {\boldsymbol{\alpha}_i^j}^T \boldsymbol{x}_0 + \beta_i^j, \tag{9}$$

and the parameters $\boldsymbol{\alpha}_i^j$, β_i^j are estimated by the maximum likelihood method. As in the case of the logistic regressions, η_i^j is an arbitrary function of $\boldsymbol{x}_0$, and the baseline class j is an arbitrary class.

3.4 Individual Logistic Regressions

Individual logistic regressions are techniques for approximating K-class multinomial logistic regressions by combining $K-1$ binary logistic regressions. As in the case of the multinomial logistic regressions, the individual logistic regressions represent the posterior probabilities p_i^* as (8), but the function η_i^j is approximated by using $K-1$ binary logistic regressions. In the following sentence, we describe the method for approximating the function η_i^j.

First, we select a class j and prepare $K-1$ binary logistic regressions π_{ij}, $i = 1, \cdots, j-1,\ j+1, \cdots, K$. The binary logistic regressions π_{ij} are trained so as to estimate the probabilities $\mu_{ij}^* = P(Y_0 = i \mid Y_0 = i \text{ or } Y_0 = j,\ \boldsymbol{x}_0)$. Namely, we prepare $K-1$ logistic regressions

$$\pi_{ij} = \frac{\exp(\eta_{ij})}{1 + \exp(\eta_{ij})} \tag{10}$$

and train the π_{ij}'s with the training set d_{ij} in (2).

The function η_{ij} in (10) can be considered as an estimate of $\log p_i^*/p_j^*$ by the following expansion:

$$\eta_{ij} = \log \frac{\pi_{ij}}{1 - \pi_{ij}} \approx \log \frac{\mu_{ij}^*}{1 - \mu_{ij}^*} = \log \frac{p_i^*}{p_j^*}, \tag{11}$$

and the function η_i^j in (8) can be also considerd as an estimate of $\log p_i^*/p_j^*$ by the following expansion:

$$\eta_i^j = \log \frac{\pi_i^j}{\pi_j^j} \approx \log \frac{p_i^*}{p_j^*}. \tag{12}$$

From this equality, replacing the function η_i^j in (8) with the function η_{ij} in (10), we can approximate the multinomial logistic regression of the baseline class j as follows:

$$\pi_i^j = \begin{cases} \dfrac{\exp(\eta_{ij})}{1 + \sum_{k \neq j}^{K} \exp(\eta_{kj})} & \text{if} \quad i \neq j \\[2ex] \dfrac{1}{1 + \sum_{k \neq j}^{K} \exp(\eta_{kj})} & \text{otherwise.} \end{cases} \tag{13}$$

4 ILR-PWC

4.1 Pattern Classification Problem of ILR-PWC

In this paper, we propose a pairwise coupling combination strategy using individual logistic regressions (ILR-PWC). As in the case of the pairwise coupling approach, the ILR-PWC approach prepares $K(K-1)/2$ binary classifiers r_{ij}, and then combines the binary classifiers together. In the ILR-PWC approach, however, logistic regression is used as the binary classifier r_{ij}, that is,

$$r_{ij} = \frac{\exp(g_{ij})}{1 + \exp(g_{ij})} \tag{14}$$

where g_{ij} is an arbitrary function of an input $\boldsymbol{x}_0$. The logistic regression r_{ij} is trained so as to estimate probability μ_{ij}^* with the training set d_{ij}. Then, the goal is to estimate the posterior probabilities p_i^* by using all r_{ij}.

To estimate the posterior probabilities, Hastie and Tibshirani [7] proposed a nonlinear and iterative algorithm, but it is difficult to investigate the accuracy. From this reason, we propose a two-stage estimation strategy. In the first stage, we construct K multinomial logistic regressions using the $K(K-1)/2$ logistic regressions r_{ij}. In the second stage, we estimate the posterior probabilities p_i^* using the K multinomial logistic regressions. In this paper, we show that the optimal estimates of p_i^* can be derived as a linear combination of the K multinomial logistic regressions, and we investigate the accuracy of our estimation strategy in comparison with individual logistic regressions.

The rest of this section is organized as follows. In section 4.2, we propose a method for constructing K multinomial logistic regressions by using individual logistic regressions. In section 4.3, we estimate the posterior probabilities p_i^* using the K multinomial logistic regressions. In section 4.4, we investigate the accuracy of the ILR-PWC approach.

4.2 Constructing Multinomial Logistic Regressions

In this section, we propose a method for constructing K multinomial logistic regressions using the $K(K-1)/2$ logistic regressions r_{ij}. First, note that r_{ij} and π_{ij} in (10) are the same estimate because they are trained so as to estimate the same probability μ_{ij}^* with the same training set d_{ij}. We can therefore approximate multinomial logistic regressions with individual logistic regressions in section 3.4. That is, we can approximate a multinomial logistic regression of a baseline class j as the following:

$$p_i^j = \begin{cases} \dfrac{\exp(g_{ij})}{1+\sum_{k\neq j}^{K}\exp(g_{kj})} & \text{if} \quad i \neq j \\ \\ \dfrac{1}{1+\sum_{k\neq j}^{K}\exp(g_{kj})} & \text{otherwise.} \end{cases} \tag{15}$$

The ILR-PWC approach prepares K multinomial logistic regressions p_i^j of the baseline class $j = 1, \cdots, K$ using (15).

4.3 Estimating Posterior Probabilities

In this section, we consider to estimate the posterior probabilities p_i^* using the K multinomial logistic regressions p_i^j. In the ILR-PWC approach, we find the estimate p_i of p_i^* so that p_i is close to the estimates $p_i^1, \cdots, p_i^K$ of the K multinomial logistic regressions. The closeness measure is the Kullback-Leibler (KL) divergence between p_i and $p_i^1, \cdots, p_i^K$. Noting further that the sum of probabilities is 1, we can write the problem of estimating p_i^* as follows:

$$\text{minimize} \quad \sum_{i=1}^{K}\sum_{j=1}^{K} p_i^j \log\frac{p_i^j}{p_i} \quad \text{subject to} \quad \sum_{i=1}^{K} p_i = 1. \tag{16}$$

In the rest of this subsection, we solve this constrained optimization problem.

We use the Lagrange multiplier method to derive the optimal estimate p_i. We first define an objective function L as follows:

$$L(p_1, \cdots, p_K, \lambda) = \sum_{i=1}^{K}\sum_{j=1}^{K} p_i^j \log\frac{p_i^j}{p_i} - \lambda\left\{\sum_{i=1}^{K} p_i - 1\right\}. \tag{17}$$

Differentiating the function L with respect to the p_i and Lagrange multiplier λ, we can obtain

$$\sum_{i=1}^{K} p_i = 1, \tag{18}$$

$$p_i = -\frac{1}{\lambda}\sum_{j=1}^{K} p_i^j. \tag{19}$$

Substituting (19) into (18), we obtain $\lambda = -K$. Further substituting $\lambda = -K$ into (19), we can derive the optimal estimate p_i as follows:

$$p_i = \frac{1}{K}\sum_{j=1}^{K} p_i^j. \tag{20}$$

Thus, we construct a multi-class classifier by using (1), (14), (15) and (20), and we call this strategy ILR-PWC (pairwise coupling combination strategy using individual logistic regressions).

4.4 Investigation of Accuracy of ILR-PWC

In this section, we compare the accuracy of the ILR-PWC approach with individual logistic regressions. Here, we use the estimation error of posterior probabilities to evaluate the accuracy of a multi-class classifier. First, we define the accuracy of the ILR-PWC approach as (21). In the same way, we define the accuracy of the individual logistic regressions as (22), but R_i^{ilr} is defined using the mean of all baseline classes since we can select an arbitrary class as the baseline class.

$$R_i^{ilr-pwc} = \mathrm{E}\left\{(p_i^* - p_i)^2\right\} \tag{21}$$

$$R_i^{ilr} = \frac{1}{K}\sum_{j=1}^{K} \mathrm{E}\left\{(p_i^* - p_i^j)^2\right\} \tag{22}$$

We can obtain (23) by transforming (21) into (24), and we can therefore show that the ILR-PWC approach is more accurate than the individual logistic regressions.

$$R_i^{ilr-pwc} \leq R_i^{ilr} \tag{23}$$

$$\begin{aligned}
\mathrm{E}\left\{(p_i^* - p_i)^2\right\} &= \mathrm{E}\left\{(p_i^* - \frac{1}{K}\sum_{j=1}^{K} p_i^j)^2\right\} \\
&= \mathrm{E}\left\{\frac{1}{K^2}(\sum_{j=1}^{K}(p_i^* - p_i^j))^2\right\} \\
&\leq \mathrm{E}\left\{\frac{1}{K}\sum_{j=1}^{K}(p_i^* - p_i^j)^2\right\}
\end{aligned} \tag{24}$$

where the last inequality is obtained by the Cauchy-Schwarz inequality.

5 Computer Simulation

We present an experimental evaluation on 7 data sets from the UCI machine learning repository [11], including glass, hayes-roth, iris, led, letter, segment and vehicle. A summary of data sets is given in Table 1. For comparison, we tested three different approaches; one-per-class (OPC), pairwise coupling (PWC) and individual logistic regressions (ILR). In our experiment, as individual binary classifiers r_{ij}, we employ feedforward neural networks with one output unit and 10 hidden units.

To evaluate our approach, we used the evaluation technique 10-fold cross-validation method, which consists of randomly dividing the data into 10 equal-sized groups and performing ten different experiments. In each run, nine of the ten groups are used to train the classifiers and the remaining group is held out for

Table 1. Experimental data set

Data Set	Entries	Attributes	Classes
glass	214	9	6
hayes-roth	132	5	3
iris	150	4	3
led	700	7	10
letter	20000	16	26
segment	2310	19	7
vehicle	846	18	4

Table 2. Average misclassification rates

dataset	OPC	PWC	ILR	ILR-PWC
glass	39.7	34.1	37.6	35.0
hayes-roth	38.0	30.4	35.6	30.4
iris	4.7	4.0	6.5	4.0
led	27.9	27.9	31.4	27.3
letter	39.8	18.6	33.3	17.5
segment	8.6	6.6	12.7	6.9
vehicle	23.2	21.2	25.3	21.3
average	26.0	20.4	26.1	20.3

the evaluation. Table 2 shows the average misclassification rates of 10 runs of 10-fold cross-validations. From Table 2, we can see that the misclassification rate of the ILR-PWC approach is better than that of the ILR approach. From Table 2, we can see that the maximal difference of misclassification rates between the PWC and ILR-PWC approach is 1.1% in letter data and the performance of the PWC and ILR-PWC approach are almost the same.

6 Conclusion

In this paper, we have focused on combining binary classifiers of pairwise coupling and have proposed a pairwise coupling combination strategy using individual logistic regressions (ILR-PWC). In particular, we have investigated the accuracy of the ILR-PWC approach, and as a result, we have shown that our combination strategy is more accurate than individual logistic regressions.

References

1. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55** (1997) 119–139
2. Cortes, C., Vapnik, V.: Support vector networks. Machine Learning **20** (1995) 273–297

3. Vapnik, V.: The nature of statistical learning theory, Springer (1995)
4. Rumelhart, D., Hinton, G., Williams, R.: Learning internal representations by error propagation. In: Rumelhart, D., McClelland, J. et al. (eds.): Parallel Distributed Processing: Volume 1: Foundations, MIT Press, Cambridge (1987) 318–362
5. Begg, C., Gray, R.: Calculation of polychotomous logistic regression parameters using individualized regressions. Biometrika **71** (1984) 11–18
6. Friedman, J.: Another approach to polychotomous classification. Technical Report, Statistics Department, Stanford University (1996)
7. Hastie, T., Tibshirani, R.: Classification by pairwise coupling. The Annals of Statistics **26** (1998) 451–471
8. Agresti, A.: Categorical Data Analysis. John Wiley & Sons (1990).
9. Hosmer, D., Lemeshow, S.: Applied logistic regression, 2nd ed. Wiley-Interscience (2000)
10. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. Annals of statistics **28** (2000) 337–374
11. Blake, C., Merz, C.: UCI repository of machine learning databases (1998)

The Novelty Detection Approach for Different Degrees of Class Imbalance

Hyoung-joo Lee and Sungzoon Cho*

Seoul National University, San 56-1, Shillim-dong, Kwanak-gu, 151-744, Seoul, Korea
imhjlee@gmail.com, zoon@snu.ac.kr

Abstract. We show that the novelty detection approach is a viable solution to the class imbalance and examine which approach is suitable for different degrees of imbalance. In experiments using SVM-based classifiers, when the imbalance is extreme, novelty detectors are more accurate than balanced and unbalanced binary classifiers. However, with a relatively moderate imbalance, balanced binary classifiers should be employed. In addition, novelty detectors are more effective when the classes have a non-symmetrical class relationship.

1 Introduction

The class imbalance refers to a situation where one class is heavily underrepresented compared to the other class in a classification problem [1]. Dealing with the class imbalance is of importance since it is not only very prevalent in various domains of problems but also a major cause for performance deterioration [2]. When one constructs a binary classifier with an imbalanced training dataset, the classifier produces lopsided outputs to the majority class. In other words, it classifies far more patterns to belong to the majority class than it should. Real world examples include fault detection in a machine, fraud detection, response modeling, and so on.

A vast number of approaches have been proposed to deal with the class imbalance [1,2,3,4,5,6]. The most popular methods try to balance the dataset with under-/over-sampling, and cost modification. A balanced binary classifier is constructed using one of the balancing methods, while a classifier is called unbalanced when no balancing method is implemented. On the other hand, the drastic solution of totally ignoring one class during training can work well for some imbalanced problems [7,8,9,10]. This approach is called novelty detection or one-class classification [11,12] where the majority class is designated as normal while the minority class as novel. A classifier learns the characteristics of the normal patterns in training data and detects novel patterns that are different from the normal ones. Geometrically speaking, a novelty detector generates a closed boundary around the normal patterns [13]. Although a novelty detector usually learns only one class, it can also learn two classes. It has been empirically shown that a novelty detector trained with a few novel patterns as well can generate a more accurate and tighter boundary [9,12].

* Corresponding author.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 21–30, 2006.

In this paper, we show that the novelty detection approach is a viable solution to the class imbalance. In particular, two types of novelty detectors, 1-SVM trained only with one class [13] and 1-SVM trained with two classes (1-SVM$_2$) [14], are compared with balanced and unbalanced SVMs. In order to investigate which approach is suitable for different degrees of class imbalance, experiments are conducted on artificial and real-world problems with varying degrees of imbalance. In the end, we examine the following conjectures:

(a) Novelty detectors are suitable for an extreme imbalance while balanced binary classifiers are suitable for a relatively moderate imbalance.
(b) A problem is called symmetrical when each class originally consists of homogeneous patterns and a classifier discriminates two classes, e.g. apples and oranges, or males and females. A problem is called non-symmetrical, when only one class is of interest and everything else belongs to another class. A classifier distinguishes one class from all other classes, e.g. apples from all other fruits. Novelty detectors are more suitable for datasets with non-symmetrical class relationships than with symmetrical relationships.
(c) As the class imbalance diminishes, a novelty detector trained with two classes improves more, compared to one trained with one class.

The following section briefly reviews the support vector-based classifiers used in this paper and Section 3 presents the experimental results. Conclusion and some remarks are given in Section 4.

2 Support Vector-Based Classifiers

Suppose a dataset $\mathbf{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ where $\mathbf{x}_i$ is a d-dimensional input pattern and y_i is its class label. Let us define the majority and the minority classes as $\mathbf{X}^+ = \{\mathbf{x}_i | y_i = +1\}$ and $\mathbf{X}^- = \{\mathbf{x}_i | y_i = -1\}$, respectively. In an imbalanced dataset, $N^+ \gg N^-$ where N^+ and N^- are the numbers of patterns in $\mathbf{X}^+$ and $\mathbf{X}^-$, respectively. We employ unbalanced SVM, balanced SVMs, 1-SVM, and 1-SVM$_2$ as listed in Table 1.

Table 1. Classifiers used: SVM indicates the standard two-class SVM. SVM-U, SVM-O, and SVM-C are balanced SVMs using under-sampling, over-sampling, and cost modification, respectively. 1-SVM and 1-SVM2 indicate one-class SVMs trained with one class and with two classes, respectively.

Unbalanced binary classifier	Balanced binary classifiers	Novelty detector with one class	Novelty detector with two classes
SVM	SVM-U SVM-O SVM-C	1-SVM	1-SVM$_2$

2.1 Support Vector Machine (SVM)

SVM finds a hyperplane that separates two classes with a maximal margin in a feature space [15]. An optimization problem can be considered:

$$\begin{aligned} \min\ & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i, \\ \text{s.t}\ & y_i(\mathbf{w}^T\mathbf{\Phi}(\mathbf{x}_i)+b) \geq 1-\xi_i, \quad \xi_i \geq 0, \quad i=1,\cdots,N, \end{aligned} \tag{1}$$

where $C \in (0,\infty]$ is the cost coefficient which controls the trade-off between the margin and the training error. The solution can be obtained by the quadratic programming techniques. In the optimal solution, only a small number of α_i's are positive where α_i's are the Lagrangian multipliers related to the training patterns. Those patterns for which α_i's are positive are called support vectors and the subset of support vectors is denoted as SV. The SVM decision function for a test pattern $\mathbf{x}$ is computed as

$$f(\mathbf{x}) = \text{sign}\Big[\mathbf{w}^T\mathbf{\Phi}(\mathbf{x})+b\Big] = \text{sign}\Bigg[\sum_{\mathbf{x}_i\in\text{SV}}\alpha_i y_i k(\mathbf{x}_i,\mathbf{x})+b\Bigg]. \tag{2}$$

2.2 Balancing with SVM

In an imbalanced problem, a typical binary classifier predict most or even all patterns to belong to the majority class [1]. Although the classification accuracy may be very high, this is not what we are interested in. We would like to construct a classifier which identifies both classes. Therefore, a balanced classifier is preferred although its accuracy may be lower than an unbalanced one. Various balancing methods have been proposed [1,2,3,4,5,6]. We apply a few of the simplest methods: under-sampling, over-sampling, and cost modification.

With under-sampling [1], N^- patterns are randomly sampled from X^+ to equate the numbers of patterns in two classes. With over-sampling [5], patterns from X^- are randomly sampled N^+ times with replacement. The two sampling methods are the most simple and the most popular. However, under-sampling may discard important information from the majority class. Over-sampling do not make additional information while increasing the number of patterns significantly. In this paper, SVM-U and SVM-O denote SVM classifiers using the under- and over-sampling methods, respectively.

For SVM, the cost modification method [3,6] is readily applicable by assigning a smaller cost to the majority class and a larger cost to the minority class to assure that the minority class is not ignored. One way to accomplish it is to modify the objective function in (1) as follows,

$$\min\ \frac{1}{2}\|\mathbf{w}\|^2 + C^+\sum_{\mathbf{x}_i\in\mathbf{X}^+}\xi_i + C^-\sum_{\mathbf{x}_i\in\mathbf{X}^-}\xi_i, \tag{3}$$

where $C^+ = \frac{N^-}{N}C$ and $C^- = \frac{N^+}{N}C$. The classifier obtained by solving (3) is denoted as SVM-C. SVM-C may lead to seriously biased results since the costs assigned entirely based on the numbers of patterns can be incorrect.

2.3 Support Vector Machine for Novelty Detection

1-SVM [13] finds a function that returns +1 for a small region containing training data and −1 for all other regions. A hyperplane $\mathbf{w}$ is defined to separate a fraction of patterns from the origin with a maximal margin in a feature space. The conventional 1-SVM performs a kind of unsupervised learning, learning only the majority class and not considering the class labels. Thus an optimization problem can be considered as follows,

$$\begin{aligned} \min \ & \frac{1}{2}\|\mathbf{w}\|^2 - \rho + \frac{1}{\nu N^+} \sum_{\mathbf{x}_i \in \mathbf{X}^+} \xi_i, \\ \text{s.t } & \mathbf{w}^T \boldsymbol{\Phi}(\mathbf{x}_i) \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad \forall \mathbf{x}_i \in \mathbf{X}^+. \end{aligned} \tag{4}$$

where $\nu \in (0,1]$ is a cost coefficient.

One can construct 1-SVM$_2$ [14] by incorporating patterns from the minority class into (4) as follows,

$$\begin{aligned} \min \ & \frac{1}{2}\|\mathbf{w}\|^2 - \rho + \frac{1}{\nu N} \sum_{i} \xi_i, \\ \text{s.t } & y_i(\mathbf{w}^T \boldsymbol{\Phi}(\mathbf{x}_i) - \rho) \geq \xi_i, \quad \xi_i \geq 0, \quad i = 1, 2, \cdots, N. \end{aligned} \tag{5}$$

Note that this is not for binary classification. The objective function is not to separate two classes but to separate the majority patterns from the origin while keeping the errors as small as possible. The solutions of (4-5) can be obtained analogously to SVM.

3 Experimental Results

The classifiers were applied to ten artificial and 24 real-world problems. For each training dataset, the degree of class imbalance varied with the fractions of the minority class being 1, 3, 5, 7, 10, 20, 30, and 40%. Each classifier was constructed based on a training dataset and evaluated on a test set which has a relatively balanced class distribution. Ten different training and test sets were randomly sampled for each problem to reduce a sampling bias.

To train the SV-based classifiers, two hyper-parameters have to be specified in advance, the RBF kernel width, σ, and the cost coefficient, C or ν. For each problem, we chose the best parameters on a hold-out dataset which has an equal number of patterns from the two classes.

3.1 Artificial Datasets

We generated five types of majority classes which reflect features such as scaling, clustering, convexity, and multi-modality. For each distribution, two types

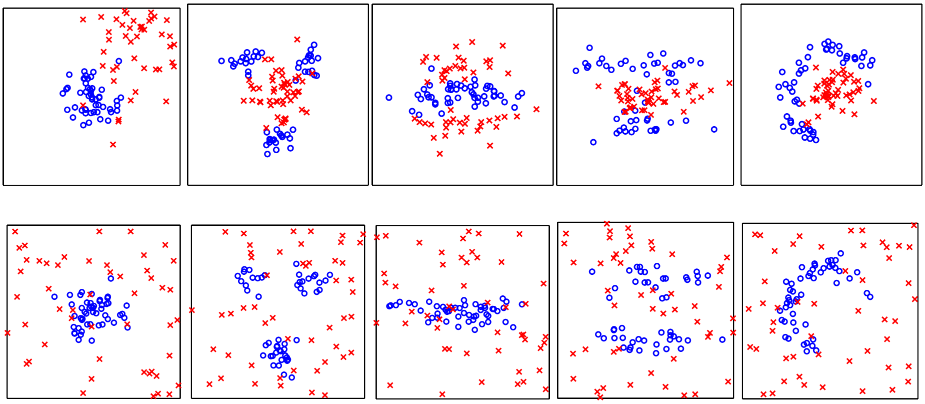
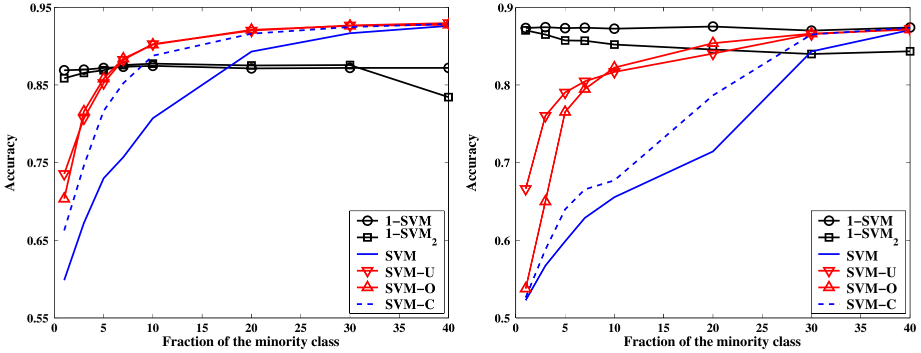


Fig. 1. The artificial datasets: The first and second rows correspond to symmetrical and non-symmetrical cases, respectively. The columns correspond to "Gauss", "Gauss3", "Ellipse", "Ellipse2", and "Horseshoe" from left to right. The circles and the crosses represent patterns from the majority and the minority classes, respectively.

of minority classes were generated. One has a multivariate Gaussian distribution while the other has the uniform distribution over the whole input space. The former corresponds to the symmetrical case while the latter to the non-symmetrical case. Thus, ten (= 5 × 2) artificial datasets were generated as shown in Fig. 1. For each dataset, 200 and 1,000 patterns were sampled from the majority class for training and test, respectively, and 1,000 patterns were sampled from the minority class for test.

Fig. 2(a) shows the average accuracies over the five symmetrical artificial datasets. When the fraction of the minority class is 5% or lower, 1-SVM and 1-SVM$_2$ are superior to the binary classifiers, balanced or not. Then, balanced classifiers, especially SVM-U and SVM-O, improved and came ahead of them as the fraction of the minority class increases. 1-SVM is generally slightly better than 1-SVM$_2$. The average accuracies over the non-symmetrical datasets are shown in Fig. 2(b). Novelty detectors are even better than in Fig. 2(a). In particular, 1-SVM is the best classifier or tied for the best for all the fractions. Unexpectedly, 1-SVM$_2$ gets gradually worse as the fraction of the minority class increases. Novelty detection is more effective for non-symmetrical datasets than for symmetrical ones. Considering that 1-SVM is better than 1-SVM$_2$, utilizing two classes does not necessarily lead to better results. As expected, unbalanced SVM did not work well and performed worst in both cases, although it caught up with the others as the fraction increased.

Fig. 3 shows examples of decision boundaries with 10% of patterns from the minority class. For the symmetrical dataset, every classifier generated a reasonable boundary. The boundaries by the binary classifiers resembled the "optimal" one. While the boundaries by the novelty detectors were different from the optimal one, they could effectively discriminate the two classes. On the other hand, for the non-symmetrical dataset, the binary classifiers failed to generate good



(a) Symmetrical

(b) Non-symmetrical

Fig. 2. The average accuracies for the artificial datasets

decision boundaries. SVM generated boundaries that will classify too large a region as the majority class. Remember that crosses can appear anywhere in the 2D space. SVM-U did its best given the dataset, but generated a boundary that was much different from the optimal one because too many patterns from the majority class were discarded. Another drawback of SVM-U is its instability. A boundary in one trial was very different from a boundary in another. Note that we present the best looking boundary in our experiments. SVM-O and SVM-C performed poorly since the patterns from the minority class were too scarce to balance the imbalance. The novelty detectors generated boundaries similar to the optimal one, though the boundaries by 1-SVM and 1-SVM$_2$ were not exactly identical.

3.2 Real-World Datasets

A total of 21 real-world datasets were selected from UCI machine learning repository[1], Data Mining Institute (DMI)[2], Rätsch's benchmark repository[3], and Tax[4] as listed in Table 2. Digit and letter recognition problems are non-symmetrical since they were formulated to distinguish one class from all others. For the digit dataset, '1' and '3' were designated in turn as the majority classes and discriminated from all other digits, respectively. For the letter dataset, 'a', 'o', and 's' were designated in turn as the majority class. Also, the pump dataset is non-symmetrical since a small non-faulty region is to be recognized in the whole input space. Therefore, six non-symmetrical problems were formulated.

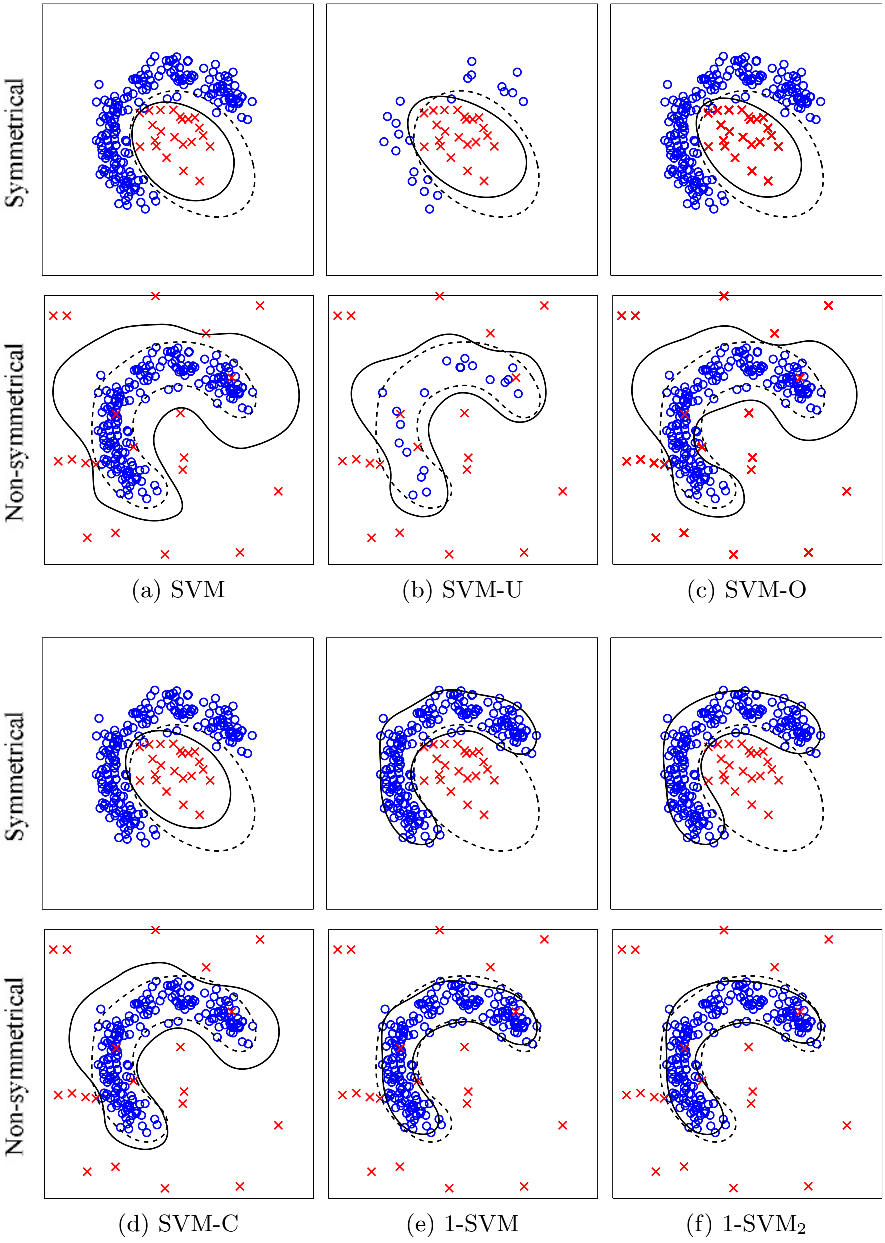
Fig. 4(a) shows the average accuracies over the 18 symmetrical real-world problems. The novelty detectors are better than the binary classifiers when the fraction is lower than 5%. Their accuracies remain still for all fractions while

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html.

[2] http://www.cs.wisc.edu/dmi/.

[3] http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm.

[4] Pump vibration datasets for fault detection used in [12]. Personal communication.



(a) SVM

(b) SVM-U

(c) SVM-O

(d) SVM-C

(e) 1-SVM

(f) 1-SVM$_2$

Fig. 3. Decision boundaries for the horseshoe dataset: Six classifiers were trained with 100 circles and ten crosses. The solid boundaries were generated by the classifiers while the broken ones are the “optimal” ones.

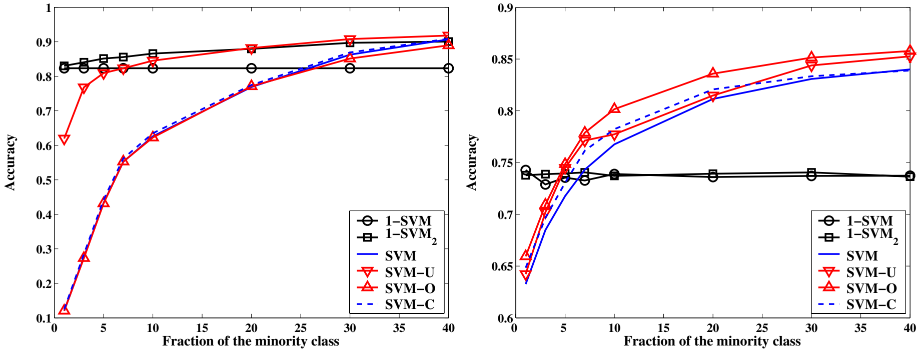
Table 2. Real-world datasets: 18 of 24 have symmetrical class distributions while three have non-symmetrical distributions

Symmetrical classes

Dataset	Source	Dataset	Source	Dataset	Source
banana	Rätsch	breast-cancer	Rätsch	bright	DMI
bupa	Rätsch	check	DMI	diabetes	Rätsch
dim	DMI	german	Rätsch	heart	Rätsch
housing	DMI	image	Rätsch	ionosphere	UCI
mush	DMI	thyroid	Rätsch	titanic	Rätsch
twonorm	Rätsch	vehicle	UCI	waveform	Rätsch

Non-symmetrical classes

Dataset	Source	Dataset	Source	Dataset	Source
digit	UCI	letter	UCI	pump	Tax



(a) Symmetrical classes

(b) Non-symmetrical classes

Fig. 4. The average accuracies for the real-world datasets

the accuracies of the binary classifiers increase steeply. When the fraction exceeds 5%, SVM-O is the best classifier. 1-SVM and 1-SVM$_2$ are equivalent to each other. Fig. 4(b) shows the average accuracies over the six non-symmetrical real-world problems. 1-SVM$_2$ is the best or tied for the best when the fraction is 20% or lower. 1-SVM$_2$ improves steadily as the fraction increases while the accuracy of 1-SVM changes little. 1-SVM is better than the binary classifiers until the fraction increases to 7%. Among the binary classifiers, SVM-U is the most accurate. The other classifiers show little difference in accuracy.

4 Conclusions and Discussion

In our experiments, the conjectures in Section 1 were investigated:

(a) With an extreme imbalance, e.g. with 5% or lower fraction of the minority class, novelty detectors are generally more accurate than binary classifiers.

On the other hand, with a moderate imbalance, e.g. with 20% or higher fraction of the minority class, balanced binary classifiers are more accurate than unbalanced binary classifier and novelty detectors. With a fraction of 5 to 20% of the minority class, the results are not conclusive.

(b) Novelty detectors perform better for the non-symmetrical problems than for the symmetrical ones, in comparison to binary classifiers. That is not surprising since solving a non-symmetrical problem is naturally fit for the novelty detection approach.

(c) The results are conflicting regarding the third conjecture. For the artificial datasets, 1-SVM$_2$ is no better than 1-SVM and its accuracy even decreases as the fraction of the minority class increases. For the real-world dataset, on the other hand, 1-SVM$_2$ is slightly better than 1-SVM. Its accuracy increases gradually for the non-symmetrical datasets. We speculate that learning only one class can be sufficient for a relatively noise-free dataset such as the artificial ones while learning two classes helps a novelty detector refine its boundary for a noisy dataset.

In summary, novelty detection approach should be considered as a candidate for imbalanced problems, especially when the imbalance is extreme. Balanced binary classifiers have comparable performances. So a balancing method should be chosen empirically depending on the problem at hand.

A few limitations have to be addressed. First, we only have considered degrees of class imbalance. There are many other factors to influence the class imbalance such as data fragmentation, complexity of data, data size to name a few [2,4]. The novelty detection approach needs to be analyzed with respect to them. Second, parameter selection was based on a balanced hold-out dataset. How to perform parameter selection with an imbalanced dataset demands further research. Third, we restricted our base classifiers to SVM in the experiments. Other families of algorithms such as neural networks and codebook-based methods need to be investigated as well.

Acknowledgement

The authors would like to thank David M.J. Tax for having provided us with the pump data and for valuable comments. This work was partially supported by grant No. R01-2005-000-103900-0 from Basic Research Program of the Korea Science and Engineering Foundation, Brain Korea 21, and Engineering Research Institute of SNU.

References

1. Kubat, M., Matwin, S.: Addressing the Curse of Imbalanced Training Sets: One-sided Selection. In: Proceedings of 14th International Conference on Machine Learning (1997) 179-186
2. Japkowicz, N., Stephen, S.: The Class Imbalance Problem: A Systematic Study. Intelligent Data Analysis 6(5) (2002) 429-450

3. Elkan, C.: The Foundations of Cost-sensitive Learning. In: Proceedings of the Seventh International Joint Conference on Artificial Intelligence (2001) 973-978
4. Weiss, G.M.: Mining with Rarity: A Unifying Framework. SIGKDD Explorations 6(1) (2004) 7-19
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE : Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16 (2002) 321-357
6. Shin, H.J., Cho, S.: Response Modeling with Support Vector Machines. Expert Systems with Applications 30(4) (2006) 746-760
7. He, C., Girolami, M., Ross, G.: Employing Optimized Combinations of One-class Classifiers for Automated Currency Validation. Pattern Recognition 37 (2004) 1085-1096
8. Japkowicz, N.: Concept-Learning in the Absence of Counter-Examples: An Autoassociation-based Approach to Classification. PhD thesis. Rutgers University, New Jersey (1999)
9. Lee, H., Cho, S.:. SOM-based Novelty Detection Using Novel Data. In: Proceedings of Sixth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL), Lecture Notes in Computer Science 3578 (2005) 359-366
10. Raskutti, B., Kowalczyk, A.: Extreme Re-balancing for SVMs: A Case Study. SIGKDD Explorations 6(1) (2004) 60-69
11. Bishop, C.: Novelty Detection and Neural Network Validation. In: Proceedings of IEE Conference on Vision, Image and Signal Processing 141(4) (1994) 217-222
12. Tax, D.M.J., Duin, R.P.W.: Support Vector Data Description. Machine Learning 54 (2004) 45-66
13. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the Support of a High-dimensional Distribution. Neural Computation 13 (2001) 1443-1471
14. Schölkopf, B., Platt, J.C., Smola, A.J.: Kernel Method for Percentile Feature Extraction. Technical Report, MSR-TR-2000-22. Microsoft Research, WA (2000)
15. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)

A Novel Multistage Classification Strategy for Handwriting Chinese Character Recognition Using Local Linear Discriminant Analysis

Lei Xu, Baihua Xiao, Chunheng Wang and Ruwei Dai

Laboratory of Complex System and Intelligent Science
Institute of Automation, Chinese Academy of Sciences
Zhongguancun East Rd, No.95, Beijing, 100080, P.R. China
lei.xu@ia.ac.cn

Abstract. In this paper we present a novel multistage classification strategy for handwriting Chinese character recognition. In training phase, we search for the most representative prototypes and divide the whole class set into several groups using prototype-based clustering. These groups are extended by nearest-neighbor rule and their centroids are used for coarse classification. In each group, we extract the most discriminative feature by local linear discriminant analysis and design the local classifier. The above-mentioned prototypes and centroids are optimized by a hierarchical learning vector quantization. In recognition phase, we first find the nearest group of the unknown sample, and then get the desired class label through the local classifier. Experiments have been implemented on CASIA database and the results show that the proposed method reaches a reasonable tradeoff between efficiency and accuracy.

1 Introduction

Handwriting Chinese character recognition (HCCR) is one of the most challenging topics in the fields of pattern recognition. There are three main factors that make HCCR difficult:

- Handwriting styles vary widely among individuals so that the boundaries between different classes are very complicated.
- The statistical features for HCCR are usually highly dimensional and the number of classes is very large.
- The actual HCCR system should possess the ability to *batch process* large amounts of documents efficiently.

To improve the classification accuracy, we should employ nonlinear classifiers with complex structure and higher-dimensional feature. However, such classifiers often result in great requirement for both computation and storage. From an application point of view, we hope to reach an acceptable tradeoff between accuracy and efficiency.

Linear discriminant analysis (LDA), as a dimension reduction technique, is usually utilized to alleviate the computation burden and speed up the recognition process. Furthermore, the criterion of LDA aims at maximizing the between-class variance while

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 31–39, 2006.

simultaneously minimizing the within-class variance, so that more accurate classification can be achieved because the samples are rearranged in the reduced feature space.

Multistage classification [1,2,3,4], as shown in Fig. 1, is another effective strategy for recognition problems on large class set. The coarse classification often utilizes *cheap* feature and structure such that less computation is required. The candidate set is selected according to the output of the coarse classifier, and consequently the most matching classifier is constructed (or selected) for fine classification. The fine classifier will perform detailed comparison and provide the final class labels.

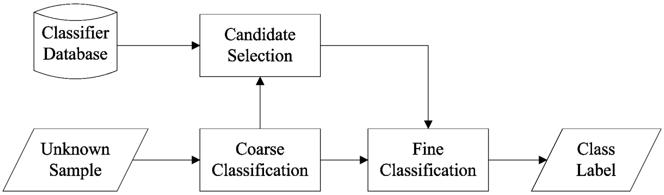


Fig. 1. Structure of multistage classification

It is natural to combine LDA and multistage classification for HCCR. In the coarse classification stage, all classes are involved, so that we perform LDA to obtain a global transformation matrix W by using all the training samples. In the fine classification stages, since only a part of classes are involved, such W isn't optimal any more and we need a local transformation matrix for each candidate set to provide locally discriminative information. However, theoretically speaking, for a system with n classes, the number of possible candidate sets is $2^n - 1$.

As is evident from the above discussion that we need an efficient candidate selection rule which can effectively exclude most of the redundant candidate sets, because we can't afford the burden of performing LDA online. Unfortunately, the existing rules, such as rank-based rule [3] and cluster-based rule [1,4], can not solve this problem.

In this paper, we present a novel multistage classification scheme for HCCR. The whole class set is divided into a set of subsets called groups, by clustering algorithms and extension rules. The adjacent groups overlap each other, and as a straightforward result, the nearest group to an unknown sample can be used as its candidate set. Since the number of groups is finite, we perform LDA for each group, so that the local discriminative features can be extracted and used for fine classification.

During the design process, we adopt a hierarchical learning vector quantization (LVQ) to improve the overall performance of our HCCR system. Firstly, we use the *global* LVQ to search for the most representative prototypes in the sense that the classification accuracy is highest. Such prototypes are used to initialize the desired groups. Secondly, we use the *group-based* LVQ to optimize the groups centroids such that the hit rate is highest. At last, after all groups have been decided, we use the *local* LVQ for each group to train the fine classifiers.

2 Group-Based Candidate Selection Rule

There are two key issues for the design of multistage classification. The first one is to improve the hit rate, which denotes the probability that the candidate set of an unknown sample's nearest region contains the correct class label. The second one is to improve the accuracy of each fine classifier. In our HCCR system, these problems are circumvented by group-based candidate selection rule and local linear discriminant analysis,respectively.

The goal of coarse classification is to decide the compact and accurate candidate set, and then construct the appropriate fine classifier. There basically exist two types of decision rules for candidate selection in the literature. Without loss of generality, we utilize the two-dimensional feature space to depict the related decision rules.

The first one is rank-based decision rule [3,5,6]. Namely, we select the classes with the highest scores based on the output of the coarse classifier and consequently construct the candidate set. An overwhelming advantage of this rule is that the fine classifier can always fix attention on the most confusable classes and exclude the irrelevant ones, as is shown in Fig. 2.

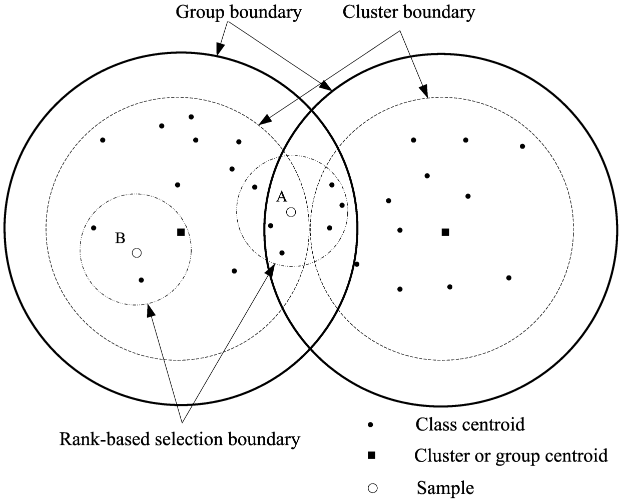


Fig. 2. Comparison of different decision rules

The second one is cluster-based decision rule [1,4]. We divide the whole class set into a number of unoverlapped clusters and utilize the cluster centroids to construct a distance-based coarse classifier. Then for an unknown sample, we choose *several* nearest clusters and use the related classes to build the candidate set. For example, sample *A* in Fig. 2 is located around the boundary between two clusters, so that both clusters should be selected to ensure high hit rate.

A common drawback of rank-based and cluster-based rule is that the number of the possible candidate sets is nearly infinite. Hence, there is little flexibility for designing an appropriate fine classifier for each group. For example, in [2], both the coarse and fine classification adopt nearest neighbor classifier. The only difference is that the fine classifier employs more samples for each class than coarse classifier.

In order to overcome this drawback, we propose the group-based decision rule. Groups are distinguished from clusters just because adjacent groups overlap each other. Such overlap is achieved by nearest-neighbor-based extension rule which will be described in the next section. For example, the boundary of the left group in Fig. 2 has been extended to the solid line. Under this circumstance, the nearest one group of an unknown sample may entirely contain its candidate set. Even though sample A is located around the boundary of the two clusters, its candidate set is still entirely contained in its nearest group (the left one).

Since the number of groups are definite, we can perform LDA for each of them, so that the most discriminative features can be extracted and the recognition rates of the fine classifiers can be greatly improved. More importantly, we can even employ different types of features and classifiers for each group.

Notes and Comments. If we excessively extend the groups, the overall hit rate will infinitely approach 100%. However, such high hit rare is meaningless because the corresponding group size will be very large. As a result, we should reach a tradeoff between the hit rate and the average size of the groups.

3 Hierarchical LVQ for Classifier Design

Considering the large class set and high-dimensional features, we employ distance-based nearest prototype classifiers (NPC(k,C)) in our HCCR system [7], where the parameter k represents the number of prototypes for each class and C is the number of classes. It is pertinent to point that our multistage classification scheme doesn't inherently prohibit other types of classifiers.

The methodology for NPC design can be found in [7,8,9]. In this paper, we employ the GLVQ algorithm due to its superior performance [7,10]. We will not describe this algorithm in detail in this paper.

3.1 Step 1: Prototype Abstraction Via Global LVQ

We first calculate the global transformation matrix W using the LDA algorithm in [11], and the rest work of step 1 and step 2 will be based on the lower-dimensional space.

The main task of step 1 is to initialize the group centroids. An intuitive choice is sample-based clustering algorithm. However, through preliminary experiments we have found that this approach suffers from slow convergence and is sensitive to the outliers in the training set. In order to overcome these drawbacks, we employ prototype-based approach. Namely, we design a NPC(K,C), and directly use the corresponding prototypes for clustering. Considering the different handwriting styles, $K > 1$ is necessary. The prototypes of each class are initialized using k-means clustering algorithm. Then the whole prototype set is trained by global LVQ algorithm, where *globe* means that the optimization process is based on the whole class set.

The NPC(K,C) designed in this step will play another role when a sample falls into the risk zone. In this condition, we utilize the cluster-based rule instead to decide the candidate set and extract the corresponding prototypes for fine classification.

3.2 Step 2: Group Extension and Group-Based LVQ

The task of step 2 is to extend the groups by nearest-neighbor (NN) rule and then optimize the centroids using supervised group-based LVQ. *Group-based* LVQ means that the objects of this optimization process are the group centroids.

To make the GLVQ algorithm meaningful, we have to define the group label of a sample before the training procedure. Since the adjacent groups overlap each other, a training sample may simultaneously belong to several groups, so that its group label is not unique.

Definition 1. *The group label Z_t of a sample (x_t, y_t) is the union of indices of the groups that contain its class label. Namely, $Z_t = \{i \,|\, y_t \in G_i\}$.*

The whole procedure for phase 2 is described as the pseudocode in Algorithm 1, where the function $NNeighbor(x, P, M)$ returns the indices of the M nearest neighbors of x in P, and G_i is the union of the indices of the classes that belong to group i. The elements in G_i are arranged in increasing order.

Algorithm 1. Pseudocode for step 2

```
Input: prototypes $P$    // $p_{(i-1)*K+1}, \cdots, p_{iK}$ belong to class $i$
Output: group $G_i$ and group centroids $g_i$,$1 \le i \le L$, where $L$
        is the number of groups
$\{g_1, \cdots, g_L\} = kmeans(P, L)$;
For $i = 1$ to $L$
    $G_i = \Phi$ (empty set);
End
For $i = 1$ to $CK$
    $q = \arg\min_{j} d(p_i, g_j)$;
    $\{b_1, \cdots, b_M\} = NNeighbor(p_i, P, M)$;
    For $j = 1$ to $M$
        $G_q = G_q \cup \{ceil(b_j/K)\}$;
    End
End
Repeat
    For $i = 1$ to $N$
        $k = \arg\min_{j \in Z_i} d(x_i, g_j)$;
        $l = \arg\min_{j \notin Z_i} d(x_i, g_j)$;
        $Update(g_k, g_l)$;      //using GLVQ algorithm [10]
    End
    $CalculateHitRate()$;
    $flag = IsConvergent()$;
Until $flag == True$
```

3.3 Step 3: Fine Classifier Design Via Local LVQ

After the groups have been decided, We should design the fine classifier for each group. Since the average size of these groups are much smaller than that of the whole class set, each group can be independently treated as a simple pattern system.

In our system, the fine classifier for each group is NPC(1). One prototype for each class is enough and can guarantee satisfying classification accuracy. We initialize the prototypes using the corresponding class means and adjust them by local LVQ, where *local* means that the optimization process is based on a subset. For each group, we repeat step 1 and obtain the local transformation matrix W_i and prototype set Q_i.

4 Recognition with Risk-Zone Rule

For an unknown sample, we first find its nearest group and then decide its class label within this group. However, as is mentioned above that we don't excessively extend each group to avoid too large group size. Therefore, if the sample is located just around the boundary of two groups, there still exist the possibility that its nearest group doesn't contain the correct class label, which will then result in incorrect classification.

To avoid such errors, we introduce the risk-zone rule. Namely, if a sample (x, y) falls into a *window*

$$min(\frac{d(x,g_k)}{d(x,g_l)}), \frac{d(x,g_k)}{d(x,g_l)}) > \eta$$

where g_k and g_l are two nearest group centroids and $0 < \eta < 1$ is a threshold, we will think that this sample is located in the risk zone. Under this circumstance, we utilize the cluster-based rule rather than the group-based one to select its candidate set. The whole algorithm for recognition is described as the pseudocode in Algorithm 2, where P_i is the subset of P that belongs to class i.

Note that the reasonable interval for η is $[0.93, 0.98]$. Although the hit rate will monotonously rise when η decreases, however, if η gets rather small, the group-based rule will degrade to the cluster-based one.

5 Experimental Results and Analysis

We conduct the experiments on a large handwriting Chinese character database collected by the Institute of Automation, Chinese Academy of Sciences. This database contains 3755 Chinese characters classes of the level 1 set of the standard GB2312-80. There are totally 300 samples for each class, and we randomly select 270 of them for training and the rest for testing. The experimental results provided in this section are all based on the test set. In preprocessing, each character image is linearly normalized to 64×64 size and then the 896-dimensional hierarchical periphery run-length (HPRL) feature is extracted.

We first perform an experiment to compare the performance of class-based rule and cluster rule. We utilize a nearest mean classifier (NMC) for coarse classification and a NPC(4,$C^{'}$) for fine classification, where $C^{'}$ is the size of the corresponding candidate set. In other words, after the candidate set is decided, we extract the prototypes that belong to the candidate set and construct the fine classifier. The experimental comparisons of these two rules can be seen from Fig. 3.

Compared with rank-based decision rule, cluster-based rule needs less computation in coarse classification, but the resulted candidate sets contain many redundant classes. On the contrary, the resulted candidate set of class-based rule is more compact and precise, so that the ultimate recognition rate is higher.

We conduct another experiment to validate the group extension method by varying the parameter M in algorithm 1. The resulted hit rate and the average group size are plotted in Fig. 4

Algorithm 2. Pseudocode for Recognition

```
Input: unknown sample $x$
Output: class label $y$
$x' = Wx$;
$\{a, b\} = NNeighbor(x', G, 2)$;    // $d(x', g_a) < d(x', g_b)$
If $\frac{d(x',g_a)}{d(x',g_b)} < \eta$
    $x'' = W_a x$;
    $q = NNeighbor(x'', Q_a, 1)$;
    $y = G_a(q)$;
Else
    $P' = \Phi$;  $G' = \Phi$;
    $\{b_1, \cdots, b_T\} = NNeighbor(x', G, T)$;
    For $i = 1$ to $T$
        $G' = G' \cup G_{b_i}$;
    End
    For $i = 1$ to $|G'|$
        $P' = P' \cup P_{G'(i)}$;
    End
    $q = NNeighbor(x', P', 1)$;
    $y = G'(ceil(q/4))$;
End
```

In application we choose $M = 33$ and then optimize the corresponding group centroids by group-based LVQ. The resulted hit rate is 98.91%. Moreover, if the risk zone rule is used with the threshold $\eta = 0.95$, the ultimate hit rate will exceed 99.47%.

The detailed results about the recognition rate and processing speed (millisecond per character) on the whole test set are listed in table 1.

From table 1 we can conclude that the proposed method with risk zone rule yields the best tradeoff between processing speed and classification accuracy. The superior recognition rate should be mainly ascribed to the local LDA which can extract the most discriminative features for the local subset. We also note that processing speed of the proposed method is much lower than that of the cluster-based method. The main reason is that the adjacent groups overlap each other so that their average size is larger. Furthermore, the second transformation before fine classification will also cost additional computation.

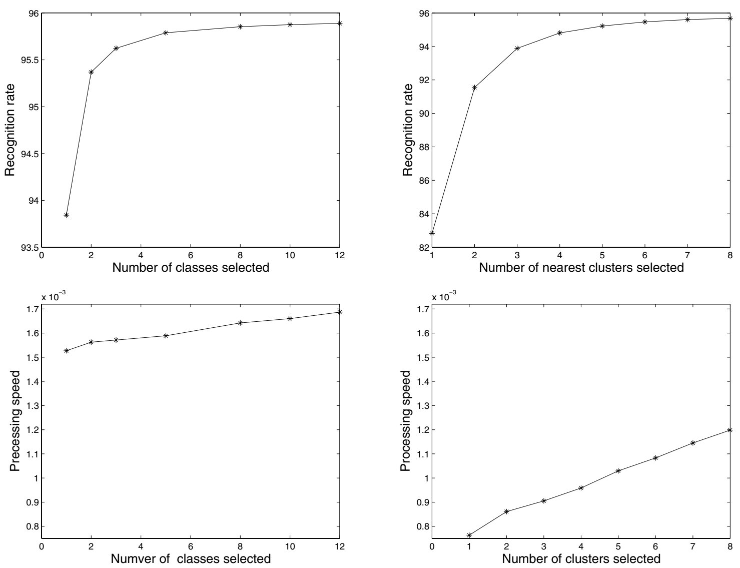


Fig. 3. The recognition rate and processing speed (seconds per character) (the left for rank-based one and the right for cluster-based one)

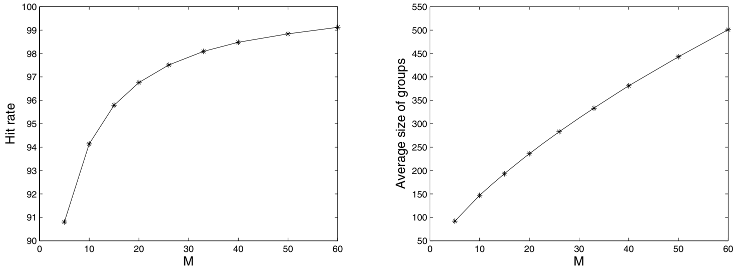


Fig. 4. The hit rate and the average group size for different M of algorithm 1

Table 1. Comparison of different methods

Classifier	Recognition rate	Processing speed
NMC	93.85	1.5269
NPC(4,3755) (trained by global GLVQ)	95.92	4.1811
class-based rule (top 8 classes selected)	95.85	1.6423
cluster-based rule (top 5 clusters selected)	95.22	**1.0297**
proposed method (without risk-zone rule)	97.30	1.8642
proposed method (with risk-zone rule,$\eta = 0.95,\ T = 3$)	**97.65**	1.7665

6 Conclusion

A novel multistage classification strategy for HCCR has been proposed in this paper. The basic idea of the proposed method is to divide the whole class set into overlapped groups such that the nearest group of a sample entirely contains its candidate set. Compared with conventional methods, our proposed method only allow one group for each unknown sample. Since the number of groups is finite, it is applicable to perform local LDA for each group. As a result, the most discriminative feature can be extracted and more accurate classification can be achieved within each group.

During the design phase, we utilize the hierarchical LVQ as a powerful tool to optimize the global prototypes, centroids and local prototypes. Experimental results show that this method can greatly improve the overall performance of our HCCR system.

Considering that the overall hit rate on test set is lower than 99%, we have introduced the risk zone rule. If samples fall into the risk zone, we use the cluster-based rule to construct the fine classifier. At the sacrifice of locally discriminative information, a large proportion of such sample can be correctly classified.

However, we find that even if the risk zone rule is utilized, the overall hit rate is still less than 99.5%. Hence, we should find more efficient group extension rules which will yield more compact groups and higher hit rate.

References

1. Liu, C.L., Mine, R., Koga, M.: Building Compact Classifier for Large Character Set Recognition Using Discriminative Feature Extraction. In: Proceedings of the Eighth International Conference on Document Analysis and Recognition. Volume 2., IEEE (2005) 846–850
2. Rodrguez, C., Soraluze, I., Muguerza, J.: Hierarchical Classifiers Based on Neighbourhood Criteria with Adaptive Computational Cost. Pattern Recognition **35**(12) (2002) 2761–2769
3. Liu, C.L., Nakagawa, M.: Precise Candidate Selection for Large Character Set Recognition by Confidence Evaluation. IEEE Transaction on Pattern Analysis and Machine Intelligence **22**(6) (2000) 636–642
4. Tseng, Y.H., Kuo, C.C., Lee, H.J.: Speeding up Chinese Character Recognition in an Automatic Document Reading System. Pattern Recognition **31**(11) (1998) 1601–1612
5. Horiuchi, T.: Class-selective Rejection Rule to Minimize the Maximum Distance between Selected Classes. Pattern Recognition **31**(10) (1998) 1579–1588
6. Ha, T.M.: The Optimum Class-selective Rejection Rule. IEEE Transaction on Pattern Analysis and Machine Intelligence **19**(6) (1997) 608–615
7. Liu, C.L., Nakagawa, M.: Evaluation of Prototype Learning Algorithms for Nearest-neighbor Classifier in Application to Handwritten Character Recognition. Pattern Recognition **34**(3) (2001) 601–615
8. Kuncheva, L.I., Bezdek, J.C.: Nearest Prototype Classification: Clustering, Genetic Algorithms, or Random Search? IEEE Transaction on Systems, Man and Cybernetics-Part C **28**(1) (1998) 160–164
9. Veenman, C.J., Reinders, M.J.T.: The Nearest Subclass Classifier: A Compromise between the Nearest Mean and Nearest Neighbor Classifier. IEEE Transaction on Pattern Analysis and Machine Intelligence **27**(9) (2005) 1417–1429
10. Sato, A., Yamada, K.: Generalized Learning Vector Quantization. In: Advances in Neural Information Processing Systems. Volume 7., MIT Press (1995) 423–429
11. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley (2000)

Prototype Based Classification Using Information Theoretic Learning

Th. Villmann[1], B. Hammer[2], F.-M. Schleif[3,4],
T. Geweniger[3,5], T. Fischer[3], and M. Cottrell[6]

[1] University Leipzig, Medical Department, Germany
[2] Clausthal University of Technology, Inst. of Computer Science, Germany
[3] University Leipzig, Inst. of Computer Science, Germany
[4] BRUKER DALTONIK Leipzig, Germany
[5] University of Applied Science Mittweida, Dep. of Computer Science, Germany
[6] University Paris I Sorbonne-Panthéon, SAMOS, France

Abstract. In this article we extend the (recently published) unsupervised information theoretic vector quantization approach based on the Cauchy–Schwarz-divergence for matching data and prototype densities to supervised learning and classification. In particular, first we generalize the unsupervised method to more general metrics instead of the Euclidean, as it was used in the original algorithm. Thereafter, we extend the model to a supervised learning method resulting in a fuzzy classification algorithm. Thereby, we allow fuzzy labels for both, data and prototypes. Finally, we transfer the idea of relevance learning for metric adaptation known from learning vector quantization to the new approach.

1 Introduction

Prototype based unsupervised vector quantization is an important task in pattern recognition. One basic advantage is the easy mapping scheme and the intuitive understanding by the concept of representative prototypes. Several methods have been established ranging from statistical approaches to neural vector quantizers [1],[2],[3]. Thereby, close connections to information theoretic learning can be drawn [4],[5],[6],[7],[8]. Based on the fundamental work of Zador, distance based vector quantization can be related to magnification in prototype base vector quantization which describes the relation between data and prototype density as a power law [9]. It can be used to design control strategies such that maximum mutual information between data and prototype density is obtained [10],[11]. However, the goal is achieved by a side effect but not directly optimized because of that distance based vector quantization methods try to minimize variants of the description error [9]. Yet, vector quantization directly optimizing information theoretic approaches become more and more important [5],[12],[8]. Two basic principles are widely used: maximization of the mutual information and minimization of the divergence, which are for uniformly distributed data

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 40–49, 2006.

equivalent. Thereby, several entropies and divergence measures exist. Shannon-entropy and Kullback-Leibler-divergence were the earliest and provided the way for the other [13],[14]. One famous entropy class is the set of Rényi's α-entropies H_α, which are a generalization of the Shannon-entropy and show interesting properties [15]. In particular, the quadratic H_2-entropy is of special interest because of its convenient properties for numerical computation. J. Principe and colleagues have been shown that, based on the Cauchy-Schwarz-inequality, a divergence measure can be derived, which, together with a consistently chosen Parzen-estimator for the densities, gives a numerically well behaved approach of information optimization based prototype based vector quantization [16].

In this contribution, we extend first this approach to more general data metrices keeping the prototype based principle. In this way a broader range of application becomes possible, for instance data equipped with only available pairwise similarity measure. Further, we allow that the similarity measure may be parametrized to obtain greater flexibility. Doing so, we are able to optimize the metric and, hence, the model with respect to these parameters, too. This strategy is known in supervised learning vector quantization as *relevance learning.* The main contribution is, that we extend the original approach to a supervised learning scheme, e.g., we transfer the ideas from the unsupervised information theoretic vector quantization to an information theoretic *learning* vector quantization approach, which is a *classification* scheme. Thereby, we allow the labels of both data and prototypes to be fuzzy resulting in a prototype based fuzzy classifier, which is an improvement in comparison to standard learning vector quantization approaches, which usually provide crisp decisions and are not able to handle fuzzy labels for data.

The paper is organized as follows: First we review the approach of information theoretic vector quantization introduced by J. Principe and colleagues, but in the more general variant of arbitrary metric. Subsequently, we explain the new model for supervised fuzzy classification scheme based on the unsupervised method and show, how relevance learning can be integrated. Numerical considerations demonstrate the abilities of the new classifying system.

2 Information Theoretic Based Vector Quantization Using the Hölder-Inequality

In the following we shortly review the derivation of a numerically well behaved divergence measure. It differs in some properties from the well-known Kullback-Leibler-divergence. However, it vanishes for identical probability densities and, therefore, it can be used in density matching optimization task like prototype based vector quantization.

Shannon's definition of entropy was extended by Rényi to a more general approach. For a given density $P(\mathbf{v})$ with data points $\mathbf{v} \in \mathbb{R}^n$, the class of differential Rényi-entropies[1] is defined as [15],[17]:

[1] We will ommit the attribute '*differential*' in the following.

$$H_{\alpha}(\rho) = \frac{1}{1-\alpha}\log\left(\int P^{\alpha}(\mathbf{v})\,d\mathbf{v}\right) \tag{1}$$

$$= \frac{1}{1-\alpha}\log V_{\alpha}(P) \tag{2}$$

for $\alpha > 0$ and $\alpha \neq 1$. The value V_{α} is denoted as *information potential.* The existing limit for $\alpha \to 1$ is the Shannon entropy

$$H(\rho) = -\int P(\mathbf{v})\log(P(\mathbf{v}))\,d\mathbf{v} \tag{3}$$

For comparison of probability density functions divergence measure are a common method. Based on Shannon entropy the Kullback-Leibler-divergence is defined as

$$KL(\rho, P) = \int \rho(\mathbf{v})\log\left(\frac{\rho(\mathbf{v})}{P(\mathbf{v})}\right)d\mathbf{v} \tag{4}$$

for given densities ρ and P. It can be generalized according to the H_{α}-entropies to

$$KL_{\alpha}(\rho, P) = \frac{1}{\alpha-1}\log\left(\int \rho(\mathbf{v})\cdot\left(\frac{\rho(\mathbf{v})}{P(\mathbf{v})}\right)^{\alpha-1}d\mathbf{v}\right). \tag{5}$$

Again, in the limit $\alpha \to 1$, $KL_{\alpha}(\rho, P) \to KL(\rho, P)$ holds. Both divergences are non-symmetric and vanish iff $\rho \equiv P$.

For investigation in practical applications of entropy computation one has to estimate and the probabilities and to replace the integral by sample mean. Thereby the most common method for density estimation is Parzen's windowing:

$$\hat{\rho}(\mathbf{v}) = \frac{1}{M\cdot\boldsymbol{\sigma}^2}\sum_{k=1}^{M} K\left(\frac{\xi(\mathbf{v}-\mathbf{w}_k)}{\boldsymbol{\sigma}^2}\right) \tag{6}$$

whereby K is a *kernel function.* $\xi(\mathbf{v}-\mathbf{w}_k)$ is assumed to be an arbitrary difference based distance measure and $\mathbf{w}_k \in \mathbb{R}^n$ are the kernel locations. In the following we will use Gauss-kernels G. Usually, both steps, Parzen estimation and sample mean, cause numerical errors. However, the sample mean error can be eliminated: Using Rényi's quadratic entropy and the properties of kernels the information potential V_2 can be estimated by

$$V_2 = \frac{1}{M^2\cdot\boldsymbol{\sigma}^4}\sum_{k=1}^{M}\sum_{j=1}^{M}\int G\left(\frac{\xi(\mathbf{v}-\mathbf{w}_k)}{\boldsymbol{\sigma}^2}\right)\cdot G\left(\frac{\xi(\mathbf{v}-\mathbf{w}_j)}{\boldsymbol{\sigma}^2}\right)d\mathbf{v} \tag{7}$$

$$= \frac{1}{M^2\cdot\boldsymbol{\sigma}^4}\sum_{k=1}^{M}\sum_{j=1}^{M} G\left(\frac{\xi(\mathbf{w}_k-\mathbf{w}_j)}{2\boldsymbol{\sigma}^2}\right) \tag{8}$$

without carrying out the integration in practice.

Unfortunately this approach can not be easily transferred to the quadratic divergence measure KL_2 because it is not quadratic according to all involved densities. Therefore, PRINCIPE suggested to use a divergence measure derived from the Cauchy-Schwarz-inequality. To do this, we first remark that the general information potential V_α in (1) defines a norm $\|\cdot\|_\alpha = (V_\alpha(\cdot))^{\frac{1}{\alpha}}$ for α-integrable functions. In particular in Hilbert-spaces the Hölder-inequality holds

$$\frac{\|\rho\|_\alpha \cdot \|P\|_{1-\alpha}}{\|\rho \cdot P\|_1} \geq 1 \tag{9}$$

with the equality iff $\rho \equiv P$ except a zero-measure set. For $\alpha = 2$ this is the Cauchy-Schwarz-inequality, which can be used for a divergence definition [8]:

$$\begin{aligned} D_{CS}(\rho, P) &= \frac{1}{2}\log\left(\int \rho^2(\mathbf{v})\, d\mathbf{v} \cdot \int P^2(\mathbf{v})\, d\mathbf{v}\right) - \log\left(\int P(\mathbf{v}) \cdot \rho(\mathbf{v})\, d\mathbf{v}\right) && (10) \\ &= \frac{1}{2}\log\left(V_2(\rho) \cdot V_2(P)\right) - \log Cr(P, \rho) && (11) \end{aligned}$$

whereby Cr is called the *cross-information potential* and D_{CS} is denoted as *Cauchy-Schwarz-divergence*. Yet, the divergence D_{CS} does not fulfill all properties of the Kullback-Leibler-divergence KL but keeping the main issue that D_{CS} vanishes for $\rho \equiv P$ (in prob.) [18]. Now we can use the entropy estimator for $V_2(\rho)$ and $V_2(P)$ according to (8) and apply the same kernel property to the cross-information potential:

$$\begin{aligned} Cr(\rho, P) &= \int P(\mathbf{v}) \cdot \rho(\mathbf{v})\, d\mathbf{v} && (12) \\ &= \frac{1}{N \cdot M \cdot \boldsymbol{\sigma}^4} \sum_{k=1}^{M} \sum_{j=1}^{N} G\left(\frac{\xi(\mathbf{v}_j - \mathbf{w}_k)}{2\sigma^2}\right) && (13) \end{aligned}$$

whereby, again, the integration is not to be carried out in practice and, hence, does not lead to numerical errors.

In (unsupervised) vector quantization the data density P is given (by samples), whereas the density ρ is the density of prototypes $\mathbf{w}_k$, which is subject of change. In information optimum vector quantization the adaptation should lead to minimization of D_{CS}.

3 Prototype Based Classification Using Cauchy-Schwarz Divergence

In the following we will extend the above outlined approach to the task of prototype based classification. Although many classification methods are known, prototype based classification is a very intuitive method. Most widely used methods are the learning vector quantization algorithms (LVQ) introduced by KOHONEN [2]. However, the adaptation dynamic does not follow a gradient of any cost function. Heuristically, the misclassification error is reduced. However, for overlapping classes the heuristic causes instabilities. Several modifications are known to overcome this problem [19],[20],[21].

From information theoretic learning point of view, an algorithm maximizing the mutual information using Re was introduced by TORKKOLA denoted as IT-LVQ [22]. However, compared to other classification approaches, this algorithm does not show convincing performance [23].

A remaining problem is that all these methods do not return fuzzy valued classification decisions as well as are not able do handle fuzzy classified data. Here we propose to use a Cauchy-Schwarz-divergence based cost function, which also can be applied to fuzzy labeled data.

Let $\mathbf{x}(\mathbf{v})$ be the fuzzy valued class label for data point $\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{y}_i$ for prototypes $\mathbf{w}_i \in \mathbb{R}^n$. Assuming, N_c is the number of possible classes, the fuzzy labels are realized as $\mathbf{x}(\mathbf{v}), \mathbf{y}_i \in \mathbb{R}^{N_c}$ with components $x_k(\mathbf{v}), y_i^k \in [0,1]$ with the normalization conditions $\sum_{k=1}^{N_c} x_k(\mathbf{v}) = 1$ and $\sum_{k=1}^{N_c} y_i^k = 1$. Let $P_{\mathbf{X}}(c)$ and $\rho_{\mathbf{Y}}(c)$ be the label density of data labels $\mathbf{X}$ and prototype labels $\mathbf{Y}$ for a given class c, respectively. We define as cost function to be minimized

$$C(\mathbf{Y},\mathbf{X}) = \sum_{c=1}^{N_c} \varpi_c \cdot 2 \cdot D_{CS}\left(\rho_{\mathbf{Y}}(\mathbf{v},c), P_{\mathbf{X}}(\mathbf{v},c)\right). \tag{14}$$

with given weighting factors ϖ_c determining the importance of a class. Because of all $P_{\mathbf{X}}(c)$ are determined by given data, minimization of $D_{CS}\left(\rho_{\mathbf{Y}}(\mathbf{v},c), P_{\mathbf{X}}(\mathbf{v},c)\right)$ is equivalent to minimization of

$$\hat{C}(\mathbf{Y},\mathbf{X}) = \sum_{c=1}^{N_c} \varpi_c \cdot \hat{C}_c(\mathbf{Y},\mathbf{X}) \tag{15}$$

with class dependent cost functions

$$\hat{C}_c(\mathbf{Y},\mathbf{X}) = \left(\log\left(V_2\left(\rho_{\mathbf{Y}}(\mathbf{v},c)\right)\right) - 2\log Cr\left(\rho_{\mathbf{Y}}(\mathbf{v},c), P_{\mathbf{X}}(\mathbf{v},c)\right)\right). \tag{16}$$

Information theoretic learning vector quantization now is taken as optimizing the prototype locations $\mathbf{w}_k$ together with their class responsibilities (labels) $\mathbf{y}^k$ according to minimization of $\hat{C}(\mathbf{Y},\mathbf{X})$.

To do so, we assume for simplicity that the variance in each data dimension is equal σ^2, the general case is straight forward. We introduce the class (label) dependent Parzen estimates

$$\hat{P}_{\mathbf{X}}(\mathbf{v},c) = \frac{1}{N}\sum_{i=1}^{N} x_c(\mathbf{v}_i) \cdot G\left(\frac{\xi(\mathbf{v}-\mathbf{v}_i)}{\boldsymbol{\sigma}^2}\right) \tag{17}$$

and

$$\hat{\rho}_{\mathbf{Y}}(\mathbf{v},c) = \frac{1}{M}\sum_{i=1}^{M} y_i^c \cdot G\left(\frac{\xi(\mathbf{v}-\mathbf{w}_i)}{\boldsymbol{\sigma}^2}\right). \tag{18}$$

We further assume for the moment that all ϖ_c are fixed and equal. Then the class dependent cost functions $\hat{C}_c(\mathbf{Y},\mathbf{X})$ can be written as

$$\hat{C}_c\left(\mathbf{Y},\mathbf{X}\right) \approx \frac{1}{2M}\sum_{i=1}^{M} y_i^c \log\left(\frac{1}{M}\sum_{j=1}^{M} y_j^c G\left(\frac{\xi\left(\mathbf{w}_i - \mathbf{w}_j\right)}{2\boldsymbol{\sigma}^2}\right)\right) \tag{19}$$

$$-\frac{1}{M}\sum_{i=1}^{M} y_i^c \log\left(\frac{1}{N}\sum_{j=1}^{N} x_c\left(\mathbf{v}_j\right)\cdot G\left(\frac{\xi\left(\mathbf{w}_i - \mathbf{v}_j\right)}{2\boldsymbol{\sigma}^2}\right)\right) \tag{20}$$

which yields the class dependent derivatives

$$\frac{\partial\hat{C}_c\left(\mathbf{Y},\mathbf{X}\right)}{\partial\mathbf{w}_k} = -\frac{1}{4\sigma^2}\left[\begin{array}{c}\dfrac{\sum_{i=1}^{M} y_i^c y_k^c G\left(\frac{\xi\left(\mathbf{w}_i,\mathbf{w}_k\right)}{2\boldsymbol{\sigma}^2}\right)\frac{\partial\xi\left(\mathbf{w}_i,\mathbf{w}_k\right)}{\partial\mathbf{w}_k}}{\sum_{i=1}^{M}\sum_{j=1}^{M} y_i^c y_j^c G\left(\frac{\xi\left(\mathbf{w}_i,\mathbf{w}_j\right)}{2\boldsymbol{\sigma}^2}\right)} \\ -\dfrac{\sum_{j=1}^{N} y_k^c x_c(\mathbf{v}_j) G\left(\frac{\xi\left(\mathbf{v}_j,\mathbf{w}_k\right)}{2\boldsymbol{\sigma}^2}\right)\frac{\partial\xi\left(\mathbf{v}_j,\mathbf{w}_k\right)}{\partial\mathbf{w}_k}}{\sum_{i=1}^{M}\sum_{j=1}^{N} y_i^c x_c(\mathbf{v}_j) G\left(\frac{\xi\left(\mathbf{v}_j,\mathbf{w}_i\right)}{2\boldsymbol{\sigma}^2}\right)}\end{array}\right] \tag{21}$$

and

$$\frac{\partial\hat{C}\left(\mathbf{Y},\mathbf{X}\right)}{\partial y_c^k} = \varpi_c \cdot \frac{\partial\hat{C}_c\left(\mathbf{Y},\mathbf{X}\right)}{\partial y_c^k} \tag{22}$$

with

$$\frac{\partial\hat{C}_c\left(\mathbf{Y},\mathbf{X}\right)}{\partial y_c^k} = \frac{\sum_{j=1}^{M} y_j^c G\left(\frac{\xi\left(\mathbf{w}_j,\mathbf{w}_k\right)}{2\boldsymbol{\sigma}^2}\right)}{\sum_{i=1}^{M}\sum_{j=1}^{M} y_i^c y_j^c G\left(\frac{\xi\left(\mathbf{w}_i,\mathbf{w}_j\right)}{2\boldsymbol{\sigma}^2}\right)} - \frac{2\sum_{j=1}^{N} x\left(\mathbf{v}_j\right) G\left(\frac{\xi\left(\mathbf{v}_j,\mathbf{w}_k\right)}{2\boldsymbol{\sigma}^2}\right)}{\sum_{i=1}^{M}\sum_{j=1}^{N} y_i^c x\left(\mathbf{v}_j\right) G\left(\frac{\xi\left(\mathbf{v}_j,\mathbf{w}_i\right)}{2\boldsymbol{\sigma}^2}\right)}. \tag{23}$$

Both gradients (21) and (23) determine the parallel stochastic gradient descent for minimization of $\hat{C}\left(\mathbf{Y},\mathbf{X}\right)$ depending on the used distance measure ξ. In case of $\xi\left(\mathbf{v}-\mathbf{w}\right)$ being the quadratic Euclidean distance, we simply have $\frac{\partial\xi(\mathbf{v}-\mathbf{w})}{\partial\mathbf{w}} = 2\left(\mathbf{v}-\mathbf{w}\right)$.

We denote the resulting adaptation algorithm

$$\begin{aligned}\triangle\mathbf{w}_k &= -\epsilon\tfrac{\partial\hat{C}(\mathbf{Y},\mathbf{X})}{\partial\mathbf{w}_k} \\ \triangle y_c^k &= -\tilde{\epsilon}\tfrac{\partial\hat{C}(\mathbf{Y},\mathbf{X})}{\partial y_c^k}\end{aligned} \tag{24}$$

as *Learning Vector Quantization based on Cauchy-Schwarz-Divergence – LVQ-CSD*

4 Applications

In a first toy example we applied the LVQ-CSD using the quadratic Euclidean distance for ξ to classify data obtained from two two-dimensional overlapping Gaussian distribution, each of them defining a data class. The overall number of data was $N = 600$ equally splitted into test and train data. We used 10 prototypes with randomly initialized positions and fuzzy labels.

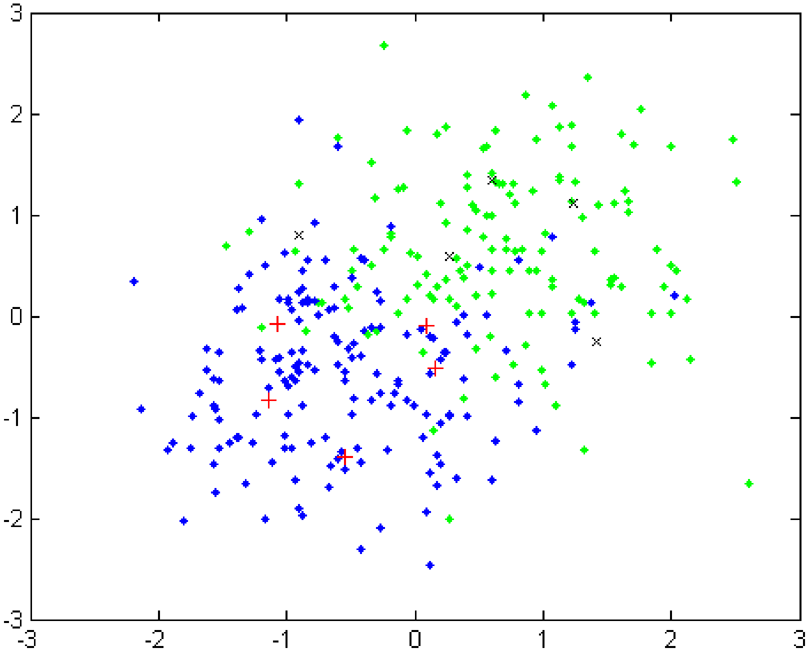


Fig. 1. Visualization of learned prototypes for LVQ-CSD in case of overlapping Gaussians, defining two classes (green, blue). The positions of prototypes are indicated by red '+' and black '×' according to their fuzzy label based majority vote for the blue and green classes, respectively.

One crucial point using Parzen estimators is the adequate choice of the kernel size $\boldsymbol{\sigma}^2$. Silverman's rule gives a rough estimation [24]. Otherwise, as pointed out in [16], $\boldsymbol{\sigma}^2$ also determines the role of cooperativeness range of prototypes in data space during adaptation, which should be larger in the beginning and smaller in the convergence phase for fine tuning. Combining both features we choose for a certain training step t

$$\boldsymbol{\sigma}(t) = \frac{3 \cdot \gamma \cdot \boldsymbol{\sigma}(0)}{1 + \delta \cdot \boldsymbol{\sigma}(0) \cdot t} \tag{25}$$

with $\gamma = 1.06 \cdot n^{-\frac{1}{5}}$ the Silverman-factor ([24]) and $\delta = 5/T$ and T being the total number of training steps. n is the data dimension and $\boldsymbol{\sigma}(0) = \boldsymbol{\sigma}$ is the original data variance.

The resulting classification accuracy (majority vote) for LVQ-CSD for the simple toy example is 93.1%, see Fig. 1. This result is comparable good to the lower accuracy obtained by standard LVQ2.1 [2], which yields 77.5%. Further, for LVQ-CSD prototypes located at overlapping border region, have balanced label vectors whereas prototypes in the center of the class regions show clear label preferences.

Table 1. Test rates for the different algorithms on the WBDC data set. For LVQ-CSD the majority vote was applied for accuracy determination.

	LVQ-CSD	LVQ2.1	GLVQ	SNG	IT-LVQ
toy sample	93.1%	77.5%	91.3%	94.9%	63.3%
PIMA	75.3%	65.3%	74.2%	78.2%	65.8%
WINE	95.5%	93.1%	98.3%	98.3%	61.9%
IONOSPHERE	69.0%	64.1%	81.4%	82.6%	56.2%

In a second mor challenging application, we investigated the behavior of the new algorithm in case of data sets from the UCI repository [25]. The data dimensions are 9, 13 and 34 for the PIMA-, the WINE- and the IONOSPHERE data, respectively. The allover number of data are 768, 178 and 351, respectively. The first and the third task are 2-class problems whereas the second one is a three-class problem. We splitted the data set for training and test randomly such that about 66% are for training.

We compare the LVQ-CSD with LVQ2.1 [2], GLVQ [20], and IT-LVQ [26] covering different principles of learning vector quantization: distance based heuristic, distance based classifier function and mutual information optimization, respectively. Because one can interpret the kernel size $\boldsymbol{\sigma}$ as a range of cooperativeness, we also added a comparison with supervised neural gas (SNG), which is an extension of GLVQ incorporating neighborhood cooperativeness [27]. The number of prototypes were chosen as 10% of train data for all algorithm, again in comparison to the earlier studies [23]. The results are depicted in Tab. 1. Except the IT-LVQ and LVQ2.1, all algorithms show comparable results with small advantages for GLVQ and, in particular SNG. LVQ-CSD shows good performance. It clearly outperforms standard LVQ2.1 and the IT-LVQ, which is based on mutual information maximization. The weak result for IONOSPHERE data set could be adressed to the well known problem arising for all Parzen estimation approaches: For high-dimensional space Parzen estimators may become insensitive because of the properties of the Euclidean norm in high-dimensional spaces: this is that according to the Euclidean distance measure most of the data lie in a thin sphere of the data space [28]. The effect could be the reason for the bad performance. Hoewever, here we have to make further investigations.

5 Conclusion and Future Work

Based on the information theoretic approach of unsupervised vector quantization by density matching using Cauchy-Schwarz-divergence, we developed a new supervised learning vector quantization algorithm, which is able to handle fuzzy labels for data as well as for prototypes. In first simulations the algorithm shows valuable results. We formulated the algorithm for general difference based distance measures $\xi(\mathbf{v}-\mathbf{w})$. However, up to now we only used the Euclidean distance. Yet, it is possible to use more complicate difference based distance measures. In particular, parametrized measures $\xi_{\boldsymbol{\lambda}}$ are of interest with parameter vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{N_{\boldsymbol{\lambda}}})$, $\lambda_i \geq 0$ and $\sum \lambda_i = 1$. Then the parametrization can be optimized for a given classification task, too. This method is known as *relevance learning* in learning vector quantization [29],[27]. For this purpose, simply the additional gradient descent $\frac{\partial \hat{C}(\mathbf{Y},\mathbf{X})}{\partial \lambda_j}$ has to be taken into account. Obviously, this idea can be transferred also to Cauchy-Schwarz-divergence as cost function of the unsupervised information theoretic vector quantization, which also would allow an adapted metric for improved performance. The analyze of these extensions in practical applications is subject of current research.

References

1. Simon Haykin, *Neural Networks - A Comprehensive Foundation*, IEEE Press, New York, 1994.
2. Teuvo Kohonen, *Self-Organizing Maps*, vol. 30 of *Springer Series in Information Sciences*, Springer, Berlin, Heidelberg, 1995, (Second Extended Edition 1997).
3. Erkki Oja and Jouko Lampinen, “Unsupervised learning for feature extraction”, in *Computational Intelligence Imitating Life*, Jacek M. Zurada, Robert J. Marks II, and Charles J. Robinson, Eds., pp. 13–22. IEEE Press, 1994.
4. R. Brause, *Neuronale Netze*, B. G. Teubner, Stuttgart, 2nd. edition, 1995.
5. G. Deco and D. Obradovic, *An Information-Theoretic Approach to Neural Computing*, Springer, Heidelberg, New York, Berlin, 1997.
6. A. K. Jain, R. P.W. Duin, and J. Mao, “Statistical pattern recognition: A review”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 4–37, 2000.
7. J.N. Kapur, *Measures of Information and their Application*, Wiley, New Delhi, 1994.
8. J. C. Principe, J.W. Fischer III, and D. Xu, “Information theoretic learning”, in *Unsupervised Adaptive Filtering*, S. Haykin, Ed. Wiley, New York, NY, 2000.
9. P. L. Zador, “Asymptotic quantization error of continuous signals and the quantization dimension”, *IEEE Transaction on Information Theory*, , no. 28, pp. 149–159, 1982.
10. Marc M. Van Hulle, *Faithful Representations and Topographic Maps*, Wiley Series and Adaptive Learning Systems for Signal Processing, Communications, and Control. Wiley & Sons, New York, 2000.
11. T. Villmann and J.-C. Claussen, “Magnification control in self-organizing maps and neural gas”, *Neural Computation*, vol. 18, no. 2, pp. 446–469, February 2006.
12. Marc M. Van Hulle, “Joint entropy maximization in kernel-based topographic maps”, *Neural Computation*, vol. 14, no. 8, pp. 1887–1906, 2002.
13. S. Kullback and R.A. Leibler, “On information and sufficiency”, *Annals of Mathematical Statistics*, vol. 22, pp. 79–86, 1951.
14. C.E. Shannon, “A mathematical theory of communication”, *Bell System Technical Journal*, vol. 27, pp. 379–432, 1948.
15. A. Renyi, “On measures of entropy and information”, in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability.* 1961, University of California Press.
16. T. Lehn-Schiler, A. Hegde, D. Erdogmus, and J.C. Principe, “Vector quantization using information theoretic concepts”, *Natural Computing*, vol. 4, no. 1, pp. 39–51, 2005.

17. A. Renyi, *Probability Theory,* North-Holland Publishing Company, Amsterdam, 1970.
18. R. Jenssen, *An Information Theoretic Approach to Machine Learning*, PhD thesis, University of Troms, Department of Physics, 2005.
19. S. Seo and K. Obermayer, "Soft learning vector quantization", *Neural Computation*, vol. 15, pp. 1589–1604, 2003.
20. A. Sato and K. Yamada, "Generalized learning vector quantization", in *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds., pp. 423–9. MIT Press, Cambridge, MA, USA, 1996.
21. S. Seo, M. Bode, and K. Obermayer, "Soft nearest prototype classification", *IEEE Transaction on Neural Networks*, vol. 14, pp. 390–398, 2003.
22. K. Torkkola, "Feature extraction by non-parametric mutual information maximization", *Journal of Machine Learning Research*, vol. 3, pp. 1415–1438, 2003.
23. T. Villmann, F.-M. Schleif, and B. Hammer, "Comparison of relevance learning vector quantization with other metric adaptive classification methods", *Neural Networks*, vol. 19, pp. in press, 2006.
24. B.W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall, 1986.
25. C.L. Blake and C.J. Merz, "UCI repository of machine learning databases", Irvine, CA: University of California, Department of Information and Computer Science, available at: http://www.ics.uci.edu/ mlearn/MLRepository.html, 1998.
26. K. Torkkola and W.M. Campbell, "Mutual information in learning feature transformations", in *Proc. Of International Conference on Machine Learning ICML'2000*, Stanford, CA, 2000.
27. B. Hammer, M. Strickert, and Th. Villmann, "Supervised neural gas with general similarity measure", *Neural Processing Letters*, vol. 21, no. 1, pp. 21–44, 2005.
28. M. Verleysen and D. François, ", in *Computational Intelligence and Bioinspired Systems, Proceedings of the 8th International Work-Conference on Artificial Neural Networks 2005 (IWANN), Barcelona*, J. Cabestany, A. Prieto, and F. S. Hernández, Eds.
29. B. Hammer and Th. Villmann, "Generalized relevance learning vector quantization", *Neural Networks*, vol. 15, no. 8-9, pp. 1059–1068, 2002.

A Modal Symbolic Classifier for Interval Data

Fabio C.D. Silva, Francisco de A.T. de Carvalho,
Renata M.C.R. de Souza, and Joyce Q. Silva

Centro de Informatica - CIn / UFPE, Av. Prof. Luiz Freire, s/n - Cidade
Universitaria, CEP: 50740-540 - Recife - PE - Brasil
{fcds, fatc, rmcrs, jqs}@cin.ufpe.br

Abstract. A modal symbolic classifier for interval data is presented. The proposed method needs a previous pre-processing step to transform interval symbolic data into modal symbolic data. The presented classifier has then as input a set of vectors of weights. In the learning step, each group is also described by a vector of weight distributions obtained through a generalization tool. The allocation step uses the squared Euclidean distance to compare two modal descriptions. To show the usefulness of this method, examples with synthetic symbolic data sets are considered.

1 Introduction

In many data analysis problems, the individuals are described by vectors of continuous-value data that are points. However, sometimes, these points to treat can not be quite localized and their positions are then imprecise. A solution is to define uncertainty zones around the imprecise points provided the acquisition system and their positions are then to estimate, for example the parameters of a regression or classification model. The concept of uncertainty zones data constitutes a generalization of interval-valued data which are quite natural in many application where they represent uncertainty on measurements (confidence interval for instance), variability (minimum and maximum temperatures during a day).

Symbolic Data Analysis (SDA) [2] is a new domain in the area of knowledge discovery and data management, related to multivariate analysis, pattern recognition and artificial intelligence. It aims to provide suitable methods (clustering, factorial techniques, decision tree, etc.) for managing aggregated data described through multi-valued variables, where there are sets of categories, intervals, or weight (probability) distributions in the cells of the data table (for more details about SDA, see www.jsda.unina2.it). A symbolic variable is defined according to its type of domain. For example, for an object, an interval variable takes an interval of $\Re$ (the set of real numbers). A symbolic modal takes, for a object, a non-negative measure (a frequency or a probability distribution or a system of weights). If this measure is specified in terms of a *histogram*, the modal variable is called *histogram variable.*

Several supervised classification tools has been extended to handle interval and modal data. Ichino et al. [6] introduced a symbolic classifier as a region

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 50–59, 2006.

oriented approach for multi-valued data. In this approach, the classes of examples are described by a region (or set of regions) obtained through the use of an approximation of a *Mutual Neighbourhood Graph* (*MNG*) and a symbolic join operator. Souza et al. [11] proposed a *MNG* approximation to reduce the complexity of the learning step without losing the classifier performance in terms of prediction accuracy. D'Oliveira et al. [4] presented a region oriented approach in which each region is defined by the convex hull of the objects belonging to a class. Ciampi et al. [3] introduced a generalization of binary decision trees to predict the class membership of symbolic data. Prudencio et al. [9] proposed a supervised classification method from symbolic data for the model selection problem. Rossi and Conan-Guez [10] have generalized Multi-Perceptrons to work with interval data. Mali and Mitra [8] extended the fuzzy radial basis function (FRBF) network to work in the domain of symbolic data. Appice et al. [1] introduced a lazy-learning approach (labeled Symbolic Objects Nearest Neighbor SO-SNN) that extends a traditional distance weighted k-Nearest Neighbor classification algorithm to interval and modal data.

In this paper, we present a modal symbolic classifier for interval data. This method assumes a previous pre-processing step to transform interval data into modal data. In the learning step, each class of items is represented by a weight distribution obtained through a generalization tool. In the the allocation step, the new items are classified using the squared Euclidean distance between modal data. Section 2 describes modal and interval symbolic data. Section 3 introduces the modal symbolic classifier based on weight distributions. Section 4 describes the evaluation experimental considering synthetic symbolic data sets. A comparative study involving the proposed classifier and the SO-SNN approach introduced by Appice et al [1] is presented. The evaluation of the performance of these classifiers is based on the accuracy prediction that is assessed in the framework of a Monte Carlo experience with 100 replications of each set. In Section 5, the concluding remarks are given.

2 Modal and Interval Symbolic Data

In classical data analysis, the items to be grouped are usually represented as a vector of quantitative or qualitative measurements where each column represents a variable. In particular, each individual takes just one single value for each variable. In practice, however, this model is too restrictive to represent complex data since to take into account variability and/or uncertainty inherent to the data, variables must assume sets of categories or intervals, possibly even with frequencies or weights.

Let $C_k, k = 1, \ldots, K$, be a class of n_k items indexed by ki $(i = 1, \ldots, n_k)$ with $C_k \cap C_{k'} = \emptyset$ if $k \neq k'$ and $\cup_{k=1}^{K} C_k = \Omega$ a training set of size $n = \sum_{k=1}^{K} n_k$. Each item ki $(i = 1, \ldots, n_k)$ is described by p symbolic variables $X_1, \ldots, X_p$. A symbolic variable X_j is an interval variable when, given an item i of C_k

$(k = 1, \ldots, K)$, $X_j(ki) = x^j_{ki} = [a^j_{ki}, b^j_{ki}] \subseteq \mathcal{A}_j$ where $\mathcal{A}_j = [a, b]$ is an interval. A symbolic variable X_j is a histogram modal variable if, given an item i of C_k $(k = 1, \ldots, K)$, $X_j(ki) = (S(ki), \mathbf{q}(ki))$ where $\mathbf{q}(ki)$ is a vector of weights defined in $S(ki)$ such that a weight $w(m)$ corresponds to each category $m \in S(ki)$. $S(ki)$ is the support of the measure $\mathbf{q}(ki)$.

Example: An individual may have as description for a symbolic variable the modal data vector $d = (0.7[60, 65[, 0.3[65, 80[)$ where the symbolic variable, which takes values in $[60, 65[$ and $[65, 80[$, is represented by a histogram (or modal) variable, where 0.7 and 0.3 are relative frequencies of the two intervals of values.

3 A Symbolic Classifier

In this section, a modal symbolic classifier for interval data is presented. Two main steps are involved in the construction of this classifier.

1. *Learning step.* Construction of a symbolic modal description for each class of items:
   - (a) *Pre-processing*: Transformation of interval symbolic data into modal symbolic data (vector of weight distribution) for each item of the training set.
   - (b) *Generalization*: Using the pre-processed items to obtain a modal description for each class.
2. *Allocation step.* Assignment of a new item to a class according to the proximity between the modal description of this item and the modal description of a class.
   - (a) *Pre-processing*: Transforming new interval data into modal data.
   - (b) *Affectation*: Computing the dissimilarity between each class and a new item.

3.1 Learning Step

This step aims to construct a modal symbolic description for each class synthesizing the information given by the items associated to this class.

Two step constitute the learning process: pre-processing and generalization.

Pre-processing. In this paper we consider a data transformation approach which the aim is to obtain modal symbolic data from interval data. So, the presented symbolic classifier has as input data vectors of weight distributions.

The variable X_j is transformed into a modal symbolic variable $\widetilde{X}_j$ in the following way [5]: $\widetilde{X}_j(ki) = \widetilde{x}^j_{ki} = (\widetilde{\mathcal{A}}_j, \mathbf{q}^j(ki))$, where $\widetilde{\mathcal{A}}_j = \{I^j_1, \ldots, I^j_{H_j}\}$ is a set of elementary intervals, $\mathbf{q}^j(ki) = (q^j_1(ki), \ldots, q^j_{H_j}(ki))$ and $q^j_h(ki)$ $(h = 1, \ldots, H_j)$ is defined as:

$$q^j_h(ki) = \frac{l(I^j_h \cap x^j_{ki})}{l(x^j_{ki})} \tag{1}$$

$l(I)$ being the length of a closed interval I.

The bounds of these elementary intervals $I^j_h\,(h = 1, \ldots, H_j)$ are obtained from the ordered bounds of the $n+1$ intervals $\{x^j_{11}, \ldots, x^j_{1n_1}, \ldots, x^j_{k1}, \ldots, x^j_{kn_k}, \ldots, x^j_{K1}, \ldots, x^j_{Kn_K}, [a, b]\}$. They have the following properties:

1. $\bigcup_{h=1}^{H_j} I^j_h = [a, b]$
2. $I^j_h \bigcap I^j_{h'} = \emptyset$ if $h \neq h'$
3. $\forall h\ \exists ki \in \Omega$ such that $I^j_h \bigcap x^j_{ki} \neq \emptyset$
4. $\forall ki\ \exists\, S^j(ki) \subset \{1, \ldots, H_j\} : \bigcup_{h \in S^j(ki)} I^j_h = x^j_{ki}$

Table 1 shows items of a training data set from two classes. Each item is described by an interval variable.

Table 1. Items described by a symbolic interval variable

Item	Interval Data (X_1)	Class
e_1	[10,30]	1
e_2	[25,35]	1
e_3	[90,130]	2
e_4	[125,140]	2

From the interval data describing the items, we create a set of elementary intervals $\widetilde{\mathcal{A}}_1 = \{I^1_1, \ldots, I^1_{H_1}\}$ as follows: at first, we take the set of values formed by every bound (lower and upper) of all the intervals associated to the items. Then, such set of bounds is sorted in a growing way. This set of elementary intervals is: $\widetilde{\mathcal{A}}_1 = \{I^1_1, I^1_2, I^1_3, I^1_4, I^1_5, I^1_6, I^1_7\}$ where $I^1_1 = [10, 25[$, $I^1_2 = [25, 30[$, $I^1_3 = [30, 35[$, $I^1_4 = [35, 90[$, $I^1_5 = [90, 125[$, $I^1_6 = [125, 130[$ and $I^1_7 = [130, 140]$.

Using the transformation approach into modal data, we got the following modal data table:

Table 2. Items described by a modal symbolic variable

Item	Modal Data ($\widetilde{X}_1$)	Class
e_1	((0.75[10,25[),(0.25[25,30[), (0.0[30,35[), (0.0[35,90[), (0.0[90,125[), (0.0[125,130[), (0.0[130,140[))	1
e_2	((0.0[10,25[),(0.50[25,30[), (0.50[30,35[), (0.0[35,90[), (0.0[90,125[), (0.0[125,130[), (0.0[130,140[))	1
e_3	((0.0[10,25[),(0.0[25,30[), (0.0[30,35[), (0.0[35,90[), (0.88[90,125[), (0.12[125,130[), (0.0[130,140[))	2
e_4	((0.0[10,25[),(0.0[25,30[), (0.0[30,35[), (0.0[35,90[), (0.0[90,125[), (0.33[125,130[), (0.67[130,140[))	2

Generalization. This step aims to represent each class as a modal symbolic example. The symbolic description of each class is a generalization of the modal symbolic description of its items.

Let C_k be a class of n_k items. Each item of C_k is represented as a vector of modal symbolic data. This class is also represented as a vector of modal symbolic data $\widetilde{\mathbf{g}}_k = (\widetilde{g}_k^1, \ldots, \widetilde{g}_k^p)$, $\widetilde{g}_k^j = (\widetilde{\mathcal{A}}_j, \mathbf{v}^j(k))$ $(j = 1, \ldots, p)$, where $\mathbf{v}^j(k) = (v_1^j(k), \ldots, v_{H_j}^j(k))$ is a vector of weights. Notice that for each variable the modal symbolic data presents the same support $\widetilde{\mathcal{A}}_j = \{I_1^j, \ldots, I_{H_j}^j\}$ for all individuals and prototypes.

The weight $v_h^j(k)$ is computed as follows:

$$v_h^j(k) = \frac{1}{n_k} \sum_{i \in C_k} q_h^j(ki) \tag{2}$$

Table 3 shows the modal symbolic description for each class of the Table 2.

Table 3. Classes described as a modal symbolic description

Class	Modal Data ($\widetilde{X}_1$)
1	((0.375[10,25[),(0.375[25,30[), (0.25[30,35[), (0.0[35,90[) (0.0[90,125[), (0.0[125,130[), (0.0[130,140[))
2	((0.0[10,25[),(0.0[25,30[), (0.0[30,35[), (0.0[35,90[) (0.44[90,125[), (0.225[125,130[), (0.335[130,140[))

3.2 Allocation Step

The allocation of a new item to a group is based on a dissimilarity function, which compares the modal description of the new item and the modal description of a class. Two steps also constitute the allocation process.

Pre-processing. Let $\mathbf{x}_\omega = (x_\omega^1 = [a_\omega^1, b_\omega^1], \ldots, x_\omega^p = [a_\omega^p, b_\omega^p])$ be the interval description of a item to be classified ω. The aim of this step is to transform the interval description of this item into a modal symbolic description.

Here, this is achieved through the following steps:

1. Update the bounds of the set of elementary intervals $\widetilde{\mathcal{A}}_j = \{I_1^j, \ldots, I_{H_j}^j\}$ considering the bounds of the interval $[a_\omega^j, b_\omega^j]$ to create the new elementary intervals $\widetilde{\mathcal{A}}_j^* = \{I_1^{*j}, \ldots, I_{H_j^*}^{*j}\}$.
2. Compute the vector of weights $\mathbf{q}^j(\omega) = (q_1^j(\omega), \ldots, q_{H_j^*}^j(\omega))$ from the new set of elementary intervals I_t^{*j} $(t = 1, \ldots, H_j^*)$ as follow:

$$q_t^j(\omega) = \frac{l(I_t^{*j} \cap x_\omega^j)}{l(x_\omega^j)} \tag{3}$$

3. Update the vector of weights $\mathbf{v}^j(k) = (v_1^j(k), \ldots, v_{H_j}^j(k))$ $(k = 1, \ldots, K)$ of C_k from the new set of elementary intervals I_t^{*j} $(t = 1, \ldots, H_j^*)$ as follow:

$$v_t^j(k) = v_h^j(k) * \frac{l(I_h^j \cap I_t^{*j})}{l(I_h^j)}) \tag{4}$$

for $h \in \{1, \ldots, H_j\} / I_h^j \cap I_t^{*j} \neq \emptyset$. Otherwise, $v_t^j(k) = 0$.

Affectation step. Let ω be a new item, which is candidate to be assigned to a class C_k $(k = 1, \ldots, K)$, and its corresponding modal description for the variable j $(j = 1, \ldots, p)$ is: $\widetilde{x}_\omega^j = (\widetilde{\mathcal{A}}_j^*, \mathbf{q}^j(\omega))$. Let $\widetilde{\mathbf{g}}_k^j = (\widetilde{\mathcal{A}}_j^*, \mathbf{v}^j(k))$ be the corresponding modal description of C_k for the variable j $(j = 1, \ldots, p)$.

Here, the comparison between two vectors of cumulative weights $\mathbf{q}^j(\omega)$ and $\mathbf{v}^j(k)$ for the variable j is accomplished by a suitable squared Euclidean distance:

$$d^2(\mathbf{q}^j(\omega), \mathbf{v}^j(k)) = \sum_{h=1}^{H_j^*} (q_h^j(\omega) - v_h^j(k))^2 \tag{5}$$

The *classification rule* is defined as follow: ω is affected to the class C_k if

$$\phi(\omega, C_k) \leq \phi_1(\omega, C_m), \forall m \in \{1, \ldots, K\} \tag{6}$$

where

$$\phi(\omega, C_k) = \sum_{j=1}^{p} d^2(\mathbf{q}^j(\omega), \mathbf{v}^j(k)) \tag{7}$$

Example: Let ω be a new item with the description $[8, 28]$ for an interval variable. Considering the modal description of the classes 1 and 2 of the Table 3, we have $\phi(\omega, C_1) = 0.2145$ and $\phi(\omega, C_2) = 1.3554$. Therefore, this new item ω will be affected to the class C_1.

4 Experimental Evaluation

In order to show the usefulness of the proposed symbolic classifier, this section presents an experimental evaluation based on prediction accuracy with two synthetic interval data sets. Our aim is to compare the modal symbolic classifier presented in this paper with the Symbolic Objects Nearest Neighbor (SO-NN) method introduced by Appice et al. [1] based on an extension of the tradicional weighted k-Nearest Neighbor classifier (k-NN) to modal and interval symbolic data.

Like the traditional classifier (k-NN), the SO-NN classifier also requires only a dissimilarity measure and a positive integer k to define the number of the neighborhood used to classifier a new item of the test set. So, Appice et al. [1]

investigated the performance of the SO-NN classifier using different dissimilarity measures for symbolic data and selecting the optimal k from the interval $[1, \sqrt{n}]$. Moreover, for modal data, they used the KT-estimate [7] to estimate the probability (or weight) distribution of modal variables when the distribution has a zero-valued probability for some categories.

In this evaluation, the accuracy of the SO-NN classifier will performed by using the squared Euclidean distance of the equation (7) and the following values to determine the neighborhood of a test item: $k = 5$, $k = 10$ and $k = 15$.

4.1 Synthetic Interval Data Sets

In each experiment, we considered two standard quantitative data sets in $\Re^2$. Each data set has 250 points scattered among three classes of unequal sizes: two classes with ellipse shapes and sizes 70 and 80 and one class with spherical shape of size 100. Each class in these quantitative data sets were drawn according to a bi-variate normal distribution with vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ represented by:

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \text{ and } \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$$

We will consider two different configurations for the standard quantitative data sets: 1) data drawn according to a bi-variate normal distribution with well separated classes and 2) data drawn according to a bi-variate normal distribution with overlapping classes.

Each data point (z_1, z_2) of each one of these synthetic quantitative data sets is a seed of a vector of intervals (rectangle): $([z_1 - \gamma_1/2, z_1 + \gamma_1/2], [z_2 - \gamma_2/2, z_2 + \gamma_2/2])$. These parameters γ_1, γ_2 are randomly selected from the same predefined interval. The intervals considered in this paper are: $[1, 10], [1, 20], [1, 30]$, $[1, 40]$ and $[1, 50]$.

Standard data set 1 was drawn according to the following parameters (configuration 1):

a) Class 1: $\mu_1 = 17$, $\mu_2 = 34$, $\sigma_1^2 = 36$, $\sigma_2^2 = 64$ and $\rho_{12} = 0.85$;
b) Class 2: $\mu_1 = 37$, $\mu_2 = 59$, $\sigma_1^2 = 25$, $\sigma_2^2 = 25$ and $\rho_{12} = 0.0$;
c) Class 3: $\mu_1 = 61$, $\mu_2 = 31$, $\sigma_1^2 = 49$, $\sigma_2^2 = 100$ and $\rho_{12} = -0.85$;

Standard data set 2 was drawn according to the following parameters (configuration 2):

a) Class 1: $\mu_1 = 8$, $\mu_2 = 5$, $\sigma_1^2 = 16$, $\sigma_2^2 = 1$ and $\rho_{12} = 0.85$;
b) Class 2: $\mu_1 = 12$, $\mu_2 = 15$, $\sigma_1^2 = 9$, $\sigma_2^2 = 9$ and $\rho_{12} = 0.0$;
c) Class 3: $\mu_1 = 18$, $\mu_2 = 7$, $\sigma_1^2 = 16$, $\sigma_2^2 = 9$ and $\rho_{12} = -0.85$;

From these configurations of standard data sets and the predefined intervals for the parameters γ_1 and γ_2, interval data sets are obtained considering two different cases of classification with overlapping classes: 1) symbolic interval data set 1 shows a moderate case of classification with class overlapping along one interval variable and 2) symbolic interval data set 2 shows a difficult case of classification with overlapping classes along two interval variables.

Figure 1 illustrates the interval data sets 1 and 2 with parameters γ_1 and γ_2 randomly selected from the interval $[1, 10]$.

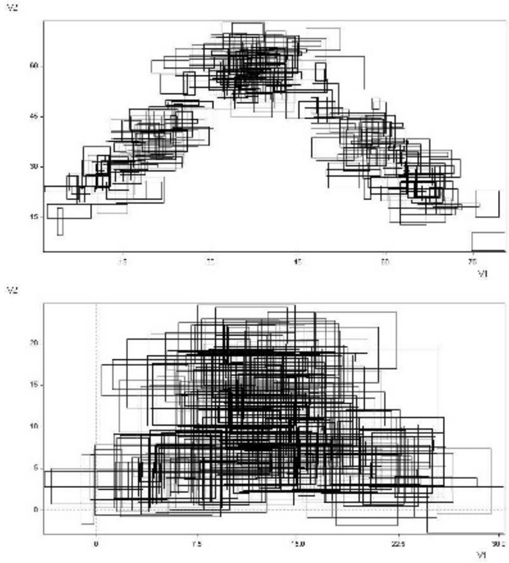


Fig. 1. Symbolic interval data sets 1 (at the top) and 2 (at the bottom)

4.2 Performance Analysis

The evaluation of these clustering methods was performed in the framework of a Monte Carlo experience: 200 replications (100 for training set and 100 for test set) are considered for each interval data set, as well as for each predefined interval. The prediction accuracy of the classifier was measured through the error rate of classification obtained from a test set. The estimated error rate of classification corresponds to the average of the error rates found between the 100 replications of the test set.

Table 1 and 2 show the values of the average and standard deviation of the error rate for the modal and SO-NN classifiers and interval data configurations 1 and 2, respectively.

For both types of interval data configurations, the average error rates for the modal classifier are less than those for the SO-NN classifier in all situations. As it is expected, the average error rates for both classifiers increase as long as the widest intervals are considered. These results show clearly that the modal classifier outperforms the SO-NN classifier.

Table 4. The average (%) and the standard deviation (in parenthesis) of the error rate for interval data set 1

Range of values	Modal Classifier	SO-NN Classifier		
		$k = 5$	$k = 10$	$k = 15$
$[1, 10]$	2.32 (0.0104)	10.90 (0.0643)	20.74 (0.0496)	31.43 (0.0296)
$[1, 20]$	2.20 (0.0101)	7.78 (0.0276)	13.48 (0.0483)	19.64 (0.0690)
$[1, 30]$	2.50 (0.0105)	8.27 (0.0267)	13.41 (0.0410)	18.56 (0.0545)
$[1, 40]$	2.78 (0.0113)	8.71 (0.0277)	14.55 (0.0443)	20.36 (0.0614)
$[1, 50]$	2.89 (0.0105)	9.65 (0.0279)	16.21 (0.0446)	22.38 (0.0569)

Table 5. The average (%) and the standard deviation (in parenthesis) of the error rate for interval data set 2

Range of values	Modal Classifier	SO-NN Classifier		
		$k = 5$	$k = 10$	$k = 15$
$[1, 10]$	9.78 (0.0184)	20.40 (0.0294)	29.48 (0.0456)	38.25 (0.0537)
$[1, 20]$	10.52 (0.0187)	25.35 (0.0373)	36.65 (0.0484)	45.76 (0.0514)
$[1, 30]$	12.68 (0.0223)	30.80 (0.0396)	44.07 (0.0467)	53.95 (0.0462)
$[1, 40]$	15.34 (0.0269)	33.74 (0.0407)	48.06 (0.0418)	58.32 (0.0365)
$[1, 50]$	18.22 (0.0322)	37.76 (0.0262)	52.52 (0.0331)	61.46 (0.0324)

5 Concluding Remarks

In this paper, a modal symbolic classifier for interval data was introduced. The proposed method needs a previous pre-processing step to transform interval symbolic data into modal symbolic data. The presented classifier has then as input a set of vectors of weights. In the learning step, each class of items is also described by a vector of weight distributions obtained through a generalization tool. The allocation step uses the squared Euclidean distance to compare the modal description of a class with the modal description of a item.

Experiments with synthetic interval data sets illustrated the usefulness of this classifier. The accuracy of the results is assessed by the error rate of classification under situations ranging from moderate to difficult cases of classification in the framework of a Monte Carlo experience. Moreover, the modal classifier is compared with the lazy SO-NN classifier for symbolic data proposed by Appice et al.

[1]. Results showed that the modal classifier proposed in this paper is superior to the SO-NN classifier in terms of error rate of classification.

Acknowledgments. The authors would like to thank CNPq (Brazilian Agency) for its financial support.

References

1. Appice, A., D'Amato, C., Esposito, F. and Malerba, D.: Classification of symbolic objects: A lazy learning approach. Intelligent Data Analysis. Special issue on Symbolic and Spatial Data Analysis: Mining Complex Data Structures. Accepted for publication, (2005)
2. Bock, H.H. and Diday, E.: Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data. Springer, Berlin Heidelberg (2000)
3. Ciampi, A, Diday, E., Lebbe, J., Perinel, E. and Vignes, R.: Growing a tree classifier with imprecise data, Pattern Recognition Leters, 21, (2000), 787-803
4. D'Oliveira, S., De Carvalho, F.A.T. and Souza, R.M.C.R.: Classification of sar images through a convex hull region oriented approach. In: N. R. Palet al. (Eds.). 11th International Conference on Neural Information Processing (ICONIP-2004), Lectures Notes in Computer Science - LNCS 3316, Springer, (2004), 769-774
5. De Carvalho., F.A.T.: Histograms in symbolic data analysis. Annals of Operations Research, 55, (1995), 299-322
6. Ichino, M., Yaguchi, H. and Diday, E.: A fuzzy symbolic pattern classifier In: Diday, E. et al (Eds.): Ordinal and Symbolic Data Analysis. Springer, Berlin, (1996), 92–102
7. Krichevsky, R.E. and Trofimov, V.K.: The performance of universal enconding. IEEE Transactions Information Theory, IT-27, (1981), 199-207
8. Mali, K. and Mitra, S.: Symbolic classification, clustering and fuzzy radial basis function network. Fyzzy sets and systems, 152, (2005), 553-564
9. Prudencio, R.B.C., Ludermir, T.B. and De Carvalho, F.A.T.: A modal symbolic classifier for selecting time series models. Pattern Recognition Leters, 25, (2004), 911-921
10. Rossi, F. and Conan-Guez, B.: Multi-layer perceptrom interval data. In: H. H. Bock (Editor). Classification, Clustering and Data Analysis (IFCS2002), Springer, (2002), 427-434
11. Souza, R.M.C.R., De Carvalho, F.A.T. and Frery, A. C.: Symbolic approach to SAR image classification. In: IEEE International Geoscience and Remote Sensing Symposium, Hamburgo, (1999), 1318–1320

Hough Transform Neural Network for Seismic Pattern Detection

Kou-Yuan Huang, Jiun-De You, Kai-Ju Chen,
Hung-Lin Lai, and An-Jin Don

Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
kyhuang2@cs.nctu.edu.tw

Abstract. Hough transform neural network is adopted to detect line pattern of direct wave and hyperbola pattern of reflection wave in a seismogram. The distance calculation from point to hyperbola is calculated from the time difference. This calculation makes the parameter learning feasible. The neural network can calculate the total error for distance from point to patterns. The parameter learning rule is derived by gradient descent method to minimize the total error. Experimental results show that line and hyperbola can be detected in both simulated and real seismic data. The network can get a fast convergence. The detection results can improve the seismic interpretation.

1 Introduction

Hough transform (HT) was used to detect parameterized shapes by mapping original image data into the parameter space [1]-[3]. The purpose of Hough transform is to find the peak value (maximum) in the parameter space. The coordinates of the peak value in parameter space is corresponding to a shape in the image space.

Seismic pattern recognition plays an important role in oil exploration. In a one-shot seismogram in Fig. 1, the travel-time curve of direct wave pattern is a straight line and the reflection wave pattern is a hyperbola. In 1985, Huang et al. had applied the Hough transform to detect direct wave (line pattern) and reflection wave (hyperbola pattern) in a one-shot seismogram [4]. However, it was not easy to determine the peak in the parameter space. Also, efficiency and memory consumption are serious drawbacks.

Neural network had been developed to solve the HT problem [5]-[7]. The Hough transform neural network (HTNN) was designed for detecting lines, circles, and ellipses [5]-[7]. But there is no application to hyperbola detection. Here, the HTNN is adopted to detect the line pattern of direct wave, and hyperbola pattern of reflection wave in a one-shot seismogram. The determination of parameters is by neural network, not by the mapping to the parameter space.

2 Proposed System and Preprocessing

Fig. 1 shows the simulated one-shot seismogram with 64 traces and every trace has 512 points. The sampling rate is 0.004 seconds. The size of the input data is 64x512. The proposed detection system is shown in Fig. 2.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 60 – 69, 2006.

The input seismogram in Fig. 1 passes through the thresholding. For seismic data, $S(x_i, t_i)$, $1 \le x_i \le 64$, $1 \le t_i \le 512$, we set a threshold *T*. For $|S(x_i, t_i)| \ge T$, data become $\mathbf{x_i} = [x_i\ t_i]^T$, *i*=1, 2, …, n. Fig. 3 is the thresholding result, where n is 252. Data are preprocessed at first, then fed into the network.

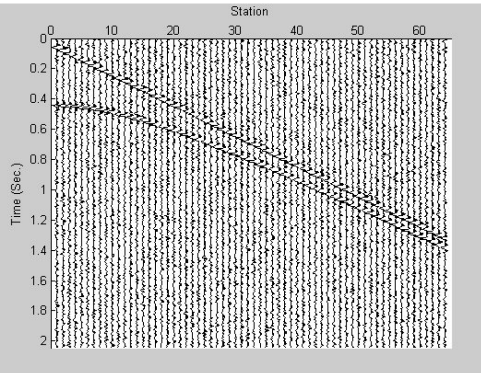


Fig. 1. One-shot seismogram

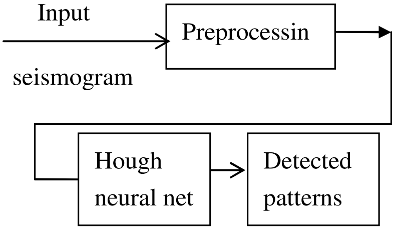


Fig. 2. Detection system for seismic patterns

3 Hough Transform Neural Network

The adopted HTNN consists of three layers: distance layer, activation function layer, and the total error layer. The network is shown in Fig. 4. It is an unsupervised network capable of detecting m parameterized objects: lines and hyperbolas, simultaneously.

Input vector $\mathbf{x_i} = [x_i\ t_i]^T$ is the *i*th point of the image, where *i*=1, 2, …, n. In the preprocessed seismic image, x_i is the trace index between shot point and receiving station, and t_i is index in time coordinate. Input each point $\mathbf{x_i}$ into distance layer, we calculate the distance $d_{ik} = D_k(\mathbf{x_i}) = D_k(x_i, t_i)$ from $\mathbf{x_i}$ to the *k*th object (line or hyperbola), *k*=1, 2, …, m. Then, d_{ik} passes through the activation function layer and the output is $Er_{ik} = 1 - f(d_{ik})$, where $f(\cdot)$ is a Gaussian basis function, i.e., $f(d_{ik}) = \exp(-\frac{d_{ik}^2}{\sigma^2})$ and Er_{ik} is the error or the modified distance of the *i*th point related to the *k*th object. Thus, when d_{ik} is near zero, Er_{ik} is also near zero. Finally, we send Er_{ik} (*k*=1, 2, …, m) to the total error layer and $Ec_i = C(\mathbf{Er}_i) = C(Er_{i1}, \ldots, Er_{ik}, \ldots, Er_{im}) = \prod_{1 \le j \le m} Er_{ij}$ is the total error of $\mathbf{x_i}$ in the network. When $\mathbf{x_i}$ is belonged to one object, then Er_{ik}=0, and Ec_i=0.

We derive the distance calculations from to point to the line detection and the hyperbola as follows.

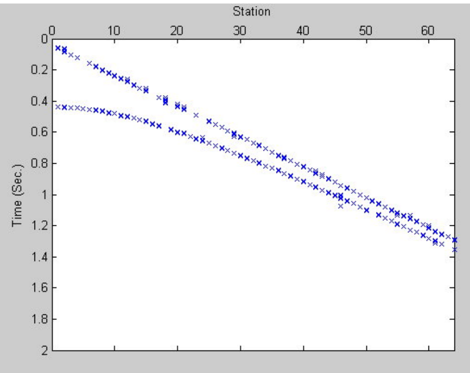


Fig. 3. Result of thresholding

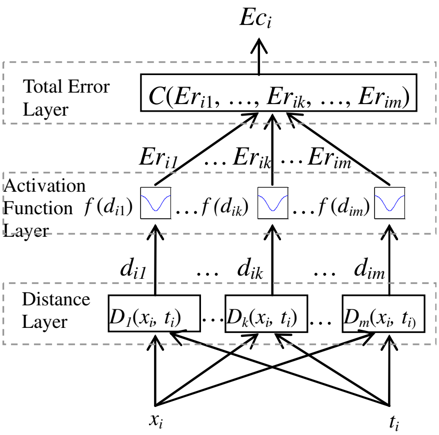

Ec_i
Total Error Layer
$C(Er_{i1}, \ldots, Er_{ik}, \ldots, Er_{im})$
Er_{i1} … Er_{ik} … Er_{im}
Activation Function Layer
$f(d_{i1})$ … $f(d_{ik})$ … $f(d_{im})$
d_{i1} … d_{ik} … d_{im}
Distance Layer
$D_1(x_i, t_i)$ … $D_k(x_i, t_i)$ … $D_m(x_i, t_i)$
x_i
t_i

Fig. 4. Hough transform neural network

3.1 Distance Layer

Distance from Point to Line. Although Basak and Das proposed Hough transform neural network to detect lines, they used the second-order equation of conoidal shapes [6]. Here, we use the direct line equation in the analysis.

For line equation $\mathrm{L_k}(\mathbf{x}) = \mathbf{w}_\mathrm{k}^\mathrm{T}\mathbf{x} + b_k = w_{k,1}x + w_{k,2}t + b_k = 0$ where $\mathbf{w}_\mathrm{k} = [w_{k,1} \quad w_{k,2}]^T$, and *k* is the *k*th line. We want to find the minimum distance from $\mathbf{x_i}$ to $\mathrm{L_k}(\mathbf{x})$. That is, minimize $\mathrm{Dist}(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}-\mathbf{x}_i\|^2$ subject to $\mathrm{L_k}(\mathbf{x}) = 0$.

From Lagrange method, the Lagrange function is $\mathrm{L}(\mathbf{x},\lambda) = \mathrm{Dist}(\mathbf{x}) + \lambda\,\mathrm{L_k}(\mathbf{x}) =$ $\frac{1}{2}\|\mathbf{x}-\mathbf{x}_i\|^2 + \lambda\left(\mathbf{w}_k^T\mathbf{x} + b_k\right)$, where λ is the Lagrange multiplier.

By the first order necessary condition, $\nabla_\mathbf{x}\, \mathrm{L}(\mathbf{x},\lambda) = (\mathbf{x}-\mathbf{x}_i) + \lambda\mathbf{w}_k = \mathbf{0}$ $\nabla_\lambda\, \mathrm{L}(\mathbf{x},\lambda) = \mathbf{w}_k^T\mathbf{x} + b_k = 0$. We obtain $\mathbf{x} = \mathbf{x}_i - \frac{\mathbf{w}_k^T\mathbf{x}_i + b_k}{\|\mathbf{w}_k\|^2}\mathbf{w}_k$. Set $\|\mathbf{w}_k\|^2 = 1$, then $\|\mathbf{x}-\mathbf{x}_i\| = \left|\mathbf{w}_k^T\mathbf{x}_i + b_k\right|$. So, for *k*th line, the output of the distance layer is

$$d_{ik} = \mathrm{D}_k(\mathbf{x}_i) = \left|\mathbf{w}_k^T\mathbf{x}_i + b_k\right| = \left|\mathrm{L}_k(\mathbf{x}_i)\right| \,. \tag{1}$$

Distance from Point to Hyperbola. In a one-shot seismogram, the reflection wave pattern is a hyperbola. The equation is $-\left(\frac{x-x_{0,k}}{a_k}\right)^2 + \left(\frac{t-t_{0,k}}{b_k}\right)^2 - 1 = 0$ and

$$\mathrm{H}_k(\mathbf{x}) = b_k \sqrt{\left(\frac{x - x_{0,k}}{a_k}\right)^2 + 1} - (t - t_{0,k}) = 0 \ . \tag{2}$$

For reflection wave in the *x-t* space, the hyperbola is on the positive *t* and positive *x* space, so b_k is positive on (2).

The true distance from point to hyperbola is complicated. Here, we consider distance in time from the point $(x_{i,}\ t_i)$ to the hyperbola $\mathrm{H}_k(\mathbf{x}) = 0$ as

$$d_{ik} = \left| b_k \sqrt{\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 + 1} - (t_i - t_{0,k}) \right| = \left|\mathrm{H}_k(\mathbf{x}_i)\right| \ . \tag{3}$$

3.2 Activation Function Layer

We use Gaussian basis function, and the activation function is defined as $Er_{ik} = f(d_{ik}) = 1 - \exp\left(-\frac{d_{ik}^2}{\sigma^2}\right)$. The Gaussian basis function controls the effect range of the point. Initially, we choose a larger σ, and σ is decreased as $\sigma / \log_2(1 + iteration)$ by iterations. This choice shows that the range of effect is decreased when the iteration number is increased.

3.3 Total Error Layer

For each point $\mathbf{x}_\mathrm{i}$ the error function is defined as $Ec_i = C(\mathbf{Er}_i) = C(Er_{i1}, \cdots, Er_{ik}, \cdots, Er_{im}) = \prod_{1 \le j \le m} Er_{ij}$. Total error is zero when the distance between input $\mathbf{x}_\mathbf{i}$ and any object is zero, i.e., Er_{ik}=0, and Ec_i=0.

4 Parametric Learning Rules

In order to minimize total error Ec_i , we use gradient descent method to adjust parameters.

The parameters of line or hyperbola can be written as a parameter vector $\mathbf{p}_k$. $\mathbf{p}_k(\mathrm{t}+1) = \mathbf{p}_k(\mathrm{t}) + \Delta\mathbf{p}_k(\mathrm{t})$ (k = 1, 2, …, m) and

$$\Delta\mathbf{p}_k = -\beta \frac{\partial Ec_i}{\partial \mathbf{p}_k}, \tag{4}$$

where β is the learning rate. From (4) and by chain rule, $\triangle\mathbf{p}_k$ can be written as

$$\begin{aligned}\Delta\mathbf{p}_k &= -\beta\left(\frac{\partial Ec_i}{\partial d_{ik}}\right)\left(\frac{\partial d_{ik}}{\partial \mathbf{p}_k}\right) = -\beta\left(\frac{\partial Ec_i}{\partial Er_{ik}}\right)\left(\frac{\partial Er_{ik}}{\partial d_{ik}}\right)\left(\frac{\partial d_{ik}}{\partial \mathbf{p}_k}\right) \\ &= -\beta\left(\frac{Ec_i}{Er_{ik}}\right)\left(\frac{2d_{ik}}{\sigma^2}\right)(1 - f(d_{ik}))\left(\frac{\partial d_{ik}}{\partial \mathbf{p}_k}\right) .\end{aligned} \tag{5}$$

We derive $\frac{\partial d_{ik}}{\partial \mathbf{p}_k}$ for line and hyperbola as follows.

4.1 Learning Rule for Line

For a line, the parameter vector is $\mathrm{p}_k = [\ w_{k,1} \quad w_{k,2} \quad b_k\ ]^T = [\ \mathrm{w}_k^T \quad b_k\ ]^T$ Thus, $\frac{\partial d_{ik}}{\partial \mathbf{p}_k} = \left[\left(\frac{\partial d_{ik}}{\partial \mathbf{w}_k}\right)^T \quad \frac{\partial d_{ik}}{\partial b_k}\right]^T = \left[\frac{\partial d_{ik}}{\partial w_{k,1}} \quad \frac{\partial d_{ik}}{\partial w_{k,2}} \quad \frac{\partial d_{ik}}{\partial b_k}\right]^T$. From (1), we can get

$$\frac{\partial d_{ik}}{\partial \mathbf{w}_k} = sign(d_{ik})\,\mathbf{x}_i \tag{6}$$

$$\frac{\partial d_{ik}}{\partial b_k} = sign(d_{ik}) \tag{7}$$

where $sign(d_{ik}) = \begin{cases} 1, & d_{ik} > 0 \\ 0, & d_{ik} = 0 \\ -1, & d_{ik} < 0 \end{cases}$. Hence, from (5), (6), and (7),

$$\Delta\mathbf{p}_k = [\Delta\mathbf{w}_k^T \quad \Delta b_k]^T = -\beta\left(\frac{Ec_i}{Er_{ik}}\right)\left(\frac{2d_{ik}}{\sigma^2}\right)(1 - f(d_{ik}))sign(d_{ik})[\mathbf{x}_i^T \ 1]^T . \tag{8}$$

Note that, in (8), $\Delta\mathbf{w}_k^T$ is proportional to $\mathbf{x}_i^T$, while Δb_k is not. That is, $\Delta\mathbf{w}_k^T$ is drastically affected by input scalar, but Δb_k is not. In order to solve this problem, we normalize the input data $[x \quad t]^T$ to satisfy

$$\mathrm{E}[\mathbf{x}] = \mathrm{E}[\mathbf{t}] = 0 \ \ and \ \ \mathrm{var}(\mathbf{x}) = \mathrm{var}(\mathbf{t}) = 1 \ . \tag{9}$$

After convergence, we obtain the parameter vector of normalized data, then we recover it to get parameter vector of original data. Without this normalization it is difficult to get the learning convergence.

4.2 Learning Rule for Hyperbola

For reflection wave, the parameter vector of hyperbola is $\mathbf{p}_k = [\ a_k \quad b_k \quad x_{0,k} \quad t_{0,k}\ ]^T$. Thus, $\frac{\partial d_{ik}}{\partial \mathbf{p}_k} = \left[\frac{\partial d_{ik}}{\partial a_k} \quad \frac{\partial d_{ik}}{\partial b_k} \quad \frac{\partial d_{ik}}{\partial x_{0,k}} \quad \frac{\partial d_{ik}}{\partial t_{0,k}}\right]^T$. From (3),

$$\begin{aligned} \frac{\partial d_{ik}}{\partial a_k} &= sign(d_{ik})\left(-\frac{b_k}{a_k}\right)\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 \Bigg/ \sqrt{\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 + 1} \\ \frac{\partial d_{ik}}{\partial b_k} &= sign(d_{ik})\sqrt{\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 + 1} \end{aligned} \tag{10}$$

$$\frac{\partial d_{ik}}{\partial x_{0,k}} = sign(d_{ik})\left(-\frac{b_k}{a_k}\right)\left(\frac{x_i - x_{0,k}}{a_k}\right) \Big/ \sqrt{\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 + 1}$$

$$\frac{\partial d_{ik}}{\partial t_{0,k}} = sign(d_{ik})$$

Then, from (5), and (10), we have

$$\Delta \mathbf{p}_k = \begin{bmatrix}\Delta a_k & \Delta b_k & \Delta x_{0,k} & \Delta t_{0,k}\end{bmatrix}^T = -\beta\left(\frac{2}{\sigma^2}\right)\left(\frac{Ec_i d_{ik}}{Er_{ik}}\right)(1 - f(d_{ik}))sign(d_{ik})$$
$$\cdot\begin{bmatrix}\left(-\frac{b_k}{a_k}\right)\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 \Big/ \sqrt{\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 + 1} \\ \sqrt{\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 + 1} \\ \left(-\frac{b_k}{a_k}\right)\left(\frac{x_i - x_{0,k}}{a_k}\right) \Big/ \sqrt{\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 + 1} \\ 1\end{bmatrix} \tag{11}$$

Also note here, input data scalar affects Δa_k, Δb_k and $\Delta x_{0,k}$. So data normalization by (9) and renormalization are also necessary for the hyperbola.

For seismic reflection wave pattern, in the geologic flat layer, we have $x_{0,k} = 0$ in (2). So the parameter vector $\mathbf{p}_k = [\ a_k \ \ b_k \ \ t_{0,k}\ ]^T$ and by (11) which implies parameter adjustment

$$\Delta \mathbf{p}_k = \begin{bmatrix}\Delta a_k & \Delta b_k & \Delta t_{0,k}\end{bmatrix}^T = -\beta\left(\frac{2}{\sigma^2}\right)\left(\frac{Ec_i d_{ik}}{Er_{ik}}\right)(1 - f(d_{ik}))sign(d_{ik})$$
$$\cdot\begin{bmatrix}\left(-\frac{b_k}{a_k}\right)\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 \Big/ \sqrt{\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 + 1} \\ \sqrt{\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 + 1} \\ 1\end{bmatrix} \tag{12}$$

Similar to [6], in the learning process, we use two stage learning and the convergence can be fast. In the first stage, we only change the bias b_k of the line in (8), and $(x_{0,k}, t_{0,k})$ of the hyperbola in (11) or $t_{0,k}$ of the hyperbola in (12) until there is no significant change in the output error. In the second stage, we adjust all parameters of line in (8) and hyperbola in (11) or (12).

The flowchart of the learning system is shown in Fig. 5. The object number m is 2, one is line and the other is hyperbola. Initially set up random parameter vectors. Then input data and adjust the parameter vector as (8) and (11) or (12). Finally, if the average error is less than a threshold, E_{th}, then the learning stops.

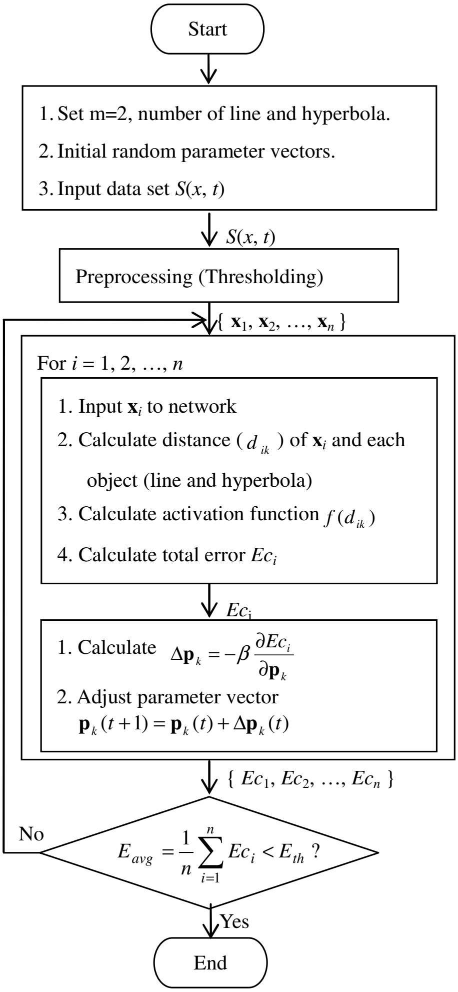

Start
1. Set m=2, number of line and hyperbola.
2. Initial random parameter vectors.
3. Input data set $S(x, t)$
$S(x, t)$
Preprocessing (Thresholding)
$\{ \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \}$
For $i = 1, 2, \ldots, n$
1. Input $\mathbf{x}_i$ to network
2. Calculate distance (d_{ik}) of $\mathbf{x}_i$ and each object (line and hyperbola)
3. Calculate activation function $f(d_{ik})$
4. Calculate total error Ec_i
Ec_i
1. Calculate $\Delta\mathbf{p}_k = -\beta \frac{\partial Ec_i}{\partial \mathbf{p}_k}$
2. Adjust parameter vector $\mathbf{p}_k(t+1) = \mathbf{p}_k(t) + \Delta\mathbf{p}_k(t)$
$\{ Ec_1, Ec_2, \ldots, Ec_n \}$
No
$E_{avg} = \frac{1}{n}\sum_{i=1}^{n} Ec_i < E_{th}$?
Yes
End

Fig. 5. Flowchart of the learning system

5 Seismic Experiments

The HTNN is applied to the simulated and real seismic data. In a simulated one-shot seismogram, the reflection layer is flat, that means $x_0 = 0$, so three parameters are detected for hyperbola. And in real seismic data, we have no prior geological knowledge, so four parameters are detected for hyperbola. In the experiments, the input data are in the image space and the results are shown in the *x-t* space.

5.1 Experiment on a Simulated One-Shot Seismogram

The image space of simulated one-shot seismogram in Fig. 1 is 64×512. After preprocessing, the input data in Fig. 6 have 252 points. Table 1 shows the detected parameters of line and hyperbola in the image space. The experimental results are shown in Fig. 6-7. We choose that β equals to 0.05, σ equals to $15/\log_2(1+iteration)$, and error threshold (E_{th}) equals to 10^{-5}. Fig. 6 shows the result of detection of direct wave and reflection wave in the *x*-*t* space. Fig. 7 shows the error versus iteration number, where the dotted line means it takes 12 iterations to change to stage two. Comparing the detection results with the original seismogram, the result of experiment is quite successful.

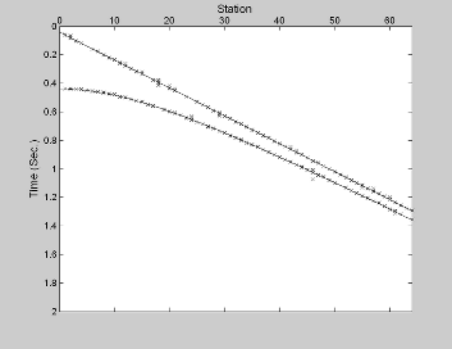


Fig. 6. Detection result: direct wave and reflection wave

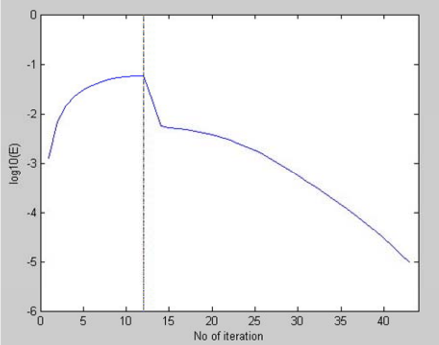


Fig. 7. Error versus iteration

5.2 Experiment on Real Seismic Data

Fig. 8 is the seismic data at Offshore Trinadad with 48 traces and 2050 points in each trace. The sampling rate is 0.004 seconds. The data are from Seismic Unix System developed by Colorado School of Mine [8]. After preprocessing, the input data in Fig. 9 have 755 points. Table 2 shows the detected parameters of line and hyperbola in the

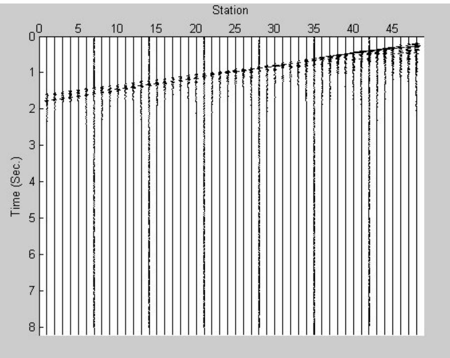


Fig. 8. Real seismic data at Offshore Trinadad

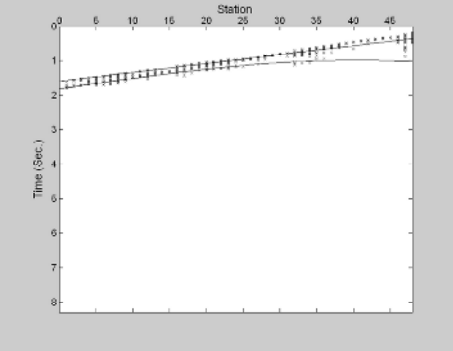


Fig. 9. Detection result: direct wave and reflection wave

image space. The results are shown in Fig. 8-10. We choose that β equals to 0.1, σ equals to $12/\log_2(1+iteration)$, and error threshold (E_{th}) equals to 2.5×10^{-4}. Fig. 9 shows the result of detection of direct wave and reflection wave in the *x-t* space. Fig. 10 shows the error versus iteration number, where the iteration number from stage one to stage two is 18.

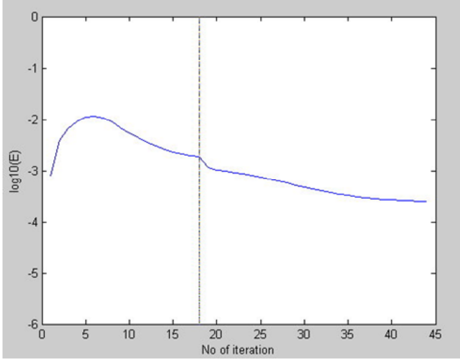


Fig. 10. Error versus iteration

Table 1. Parameters of line and hyperbola in Fig. 6 in the image space, 64×512

	w_1	w_2	b
Line	-0.040031	0.0079809	-0.082842
	a	b	t_0
Hyperbola	-21.176	-10.383	4.5614

Table 2. Parameters of line and hyperbola in Fig. 9 in the image space, 48×2050

	w_1	w_2		b
Line	0.04443	0.0068985		-2.7525
	a	b	x_0	t_0
Hyperbola	-28.371	-16.823	40.361	-37.19

6 Conclusions

HTNN is adopted to detect line pattern of direct wave and hyperbola pattern of reflection wave in a seismogram. The parameter learning rule is derived by gradient descent method to minimize the error. We use the direct line equation in the distance calculation from point to line. Also we define the vertical time difference as the distance from point to hyperbola that makes the learning feasible. In experiments, we get fast convergence in simulated data because three parameters are considered in the hyperbola detection. In real data, four parameters are in the hyperbola detection. There is no prior geological information, the detection result in line is good, but not in hyperbola. There may be 3 reasons: (1) input points are not many enough, (2) two objects are too close and affect each other, (3) there are reflections of deeper layers and affect the detection of the first layer reflection. Surely the detection results can provide a reference and improve seismic interpretation.

The result of preprocessing is quite critical for the input-output relation. More wavelet, envelope, and deconvolution processing may be needed in the preprocessing to improve the detection result.

Acknowledgments

This work was supported in part by the National Science Council, Taiwan, under NSC-94-2213-E-009-133. The authors would like to thank the use of Seismic Unix System developed by Colorado School of Mine.

References

1. Hough, P.V.C.: Method and Means for Recognizing Complex Patterns. U.S. Patent 3069654. (1962)
2. Duda, R.O., Hart, P.E.: Use of the Hough Transform to Detect Lines and Curves in Pictures. Comm. Assoc. Comput. Mach., Vol. 15. (1972) 11-15
3. Leavers, V.F.: Survey: Which Hough Transform. Computer Vision, Graphics, and Image Processing, Vol. 58, No. 2. (1993) 250-264
4. Huang, K.Y., Fu, K.S., Sheen, T.H., Cheng, S.W.: Image Processing of Seismograms: (A) Hough Transformation for the Detection of Seismic Patterns. (B) Thinning Processing in the Seismogram. Pattern Recognition, Vol. 18, No.6. (1985) 429-440
5. Dempsey, G.L., McVey, E.S.: A Hough Transform System Based on Neural Networks. IEEE Trans. Ind. Electron, Vol. 39. (1992) 522-528
6. Basak, J., Das, A.: Hough Transform Networks: Learning Conoidal Structures in a Connectionist Framework. IEEE Trans. on Neural Networks, Vol. 13, No. 2. (2002) 381-392
7. Basak, J., Das, A.: Hough Transform Network: A Class of Networks for Identifying Parametric structures. Neurocomputing, Vol. 51. (2003) 125-145
8. Yilmaz, O.: Seismic Data Processing. The Society of Exploration Geophysicists, Tulsa, (1987)

Autonomous and Deterministic Clustering for Evidence-Theoretic Classifier

Chen Li Poh[1], Loo Chu Kiong[2], and M.V.C. Rao[3]

Faculty of Engineering and Technology, Multimedia University, Jalan
Ayer Keroh Lama, Bukit Beruang, 75450 Melaka, Malaysia
lpchen@mmu.edu.my, ckloo@mmu.edu.my,
machavaram.venkata@mmu.edu.my

Abstract. This paper describes an evidence-theoretic classifier which employs global k-means algorithm as the clustering method. The classifier is based on the Dempster-Shafer rule of evidence in the form of Basic Belief Assignment (BBA). This theory combines the evidence obtained from the reference patterns to yield a new BBA. Global k-means is selected as the clustering algorithm as it can overcomes the limitation on k-means clustering algorithm whose performance depends heavily on initial starting conditions selected randomly and requires the number of clusters to be specified before using the algorithm. By testing the classifier on the medical diagnosis benchmark data, iris data and Westland vibration data, one can conclude classifier that uses global k-means clustering algorithm has higher accuracy when compared to the classifier that uses k-means clustering algorithm.

Keywords: Dempster-shafer theory, clustering, k-means algorithm.

1 Introduction

Since 1976, evidence theory has been gaining increasing acceptance in the field of artificial intelligence, particularly in the design of expert systems. Many researches have been carried out based on Dempster-Shafer theory. In [10], Dempster-Shafer theory has been applied on sensor fusion due to its ability of uncertainty management and interference mechanisms being analogous to human reasoning process. In [11], Dempster-Shafer algorithm also used to combine multi-scale data. From Thierry Denoueux work [7], a classification procedure based on the D-S theory using the k-nearest neighbor rule is proposed. Later from [4], a neural network classifier based on Dempster-Shafer theory has been presented. The research concludes that this method has exhibited excellent performance in several classification tasks and shows extremely robust performance to strong changes in the distribution of input data.

In this paper, this evidence-theoretic classifier [4] which based on Dempster-Shafer theory is tested using the medical diagnosis benchmark data, iris data and Westland Vibration data. This classifier consists of an input layer, two hidden layers and an output layer. A set of training data is fed into the classifier for learning purpose. The

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 70 – 79, 2006.

clustering of the training data is required to reduce complexity due to large amount of data and this results in faster classification and lower storage requirement. Clustering algorithm can be considered as the most important unsupervised learning problem, which similar sets of data are partitioned into homogeneous group, or clusters. A cluster is therefore a collection of objects which are similar between them and are dissimilar to the objects belonging to other clusters. Cluster membership may be defined by computing the distance between data point and cluster centers. K-means clustering [12] is one of the widely used clustering methods due to its robustness. However, this clustering algorithm has some drawbacks, influenced by the random initialization of cluster centers, which result in the convergence to a local minimum. In addition, this classifier needs to have the number of clusters to be specified before running the algorithm. In [8] and [9], research has been done on the initialization conditions to improve this algorithm. In [3], a new clustering method known as the global clustering algorithm has been presented to overcome the drawbacks from k-means clustering algorithm. This clustering method is a deterministic global clustering algorithm by using k-means algorithm as the local search procedure and proceeds in an incremental way attempting to optimally add one new cluster center at each stage instead of selecting initial cluster centers randomly. Therefore, this method is not sensitive to any initial starting points. Due to this, the evidence theoretic classifier is modified by using global k-means clustering algorithm as the clustering method for training data. The performance of the classifier by using k-means algorithm and global k-means algorithm is then evaluated. The Euclidean distance between the test vector (input pattern) and the cluster center of each prototype is computed and acts as an evidence for classification purpose. This evidence which is presented in the form of basic belief assignment (BBA) is then combined using Dempster-Shafer theory.

The paper is organized as follows: Section 2 describes the limitation for k-means algorithm and describes the general step for global k-means algorithm that is later used in evidence theoretic classifier as the clustering algorithm. Section 3 shows the evidence theoretic classifier based on Dempster-Shafer theory. The results by using the classifier on medical diagnosis benchmark data, iris data and Westland vibration data are discussed in section 4. Finally section 5 concludes the paper.

2 Global K-Means Algorithm

Data clustering is defined as the process of partitioning a given set of data into homogeneous group. K-means algorithm is one of the most popular clustering methods. It is an algorithm for clustering objects based on attributes into K disjoint subsets. The basic idea for clustering algorithm is to optimize the clustering criterion. The clustering criterion in k-means algorithm is related to the minimization of the clustering error, which is the sum of the Euclidean distances between each element and the cluster center.

K-means algorithm starts by partitioning the input data into k initial partitions by selecting the initial cluster center randomly. The membership for the patterns is decided by assigning the pattern to its nearest cluster center and a new partition can be constructed depending on this distance. New cluster center is obtained from this new partition. The clustering criterion is given by

$$E(c_1,....,c_M) = \sum_{i=1}^{M} \sum_{j=1}^{M_i} \left\| c_{ij} - \overline{c_i} \right\| \tag{1}$$

where $\overline{c_i}$ is the cluster center for the ith cluster. The algorithm will be repeated until the convergence achieved, where there is no more changing on the cluster center.

However, the convergence is not guaranteed to yield a global optimum as the performance of k-means algorithm is heavily depending on the initial positions of the cluster center that is selected randomly. The sensitivity to initial positions of the cluster centers requires several runs with different in initial positions to be scheduled in order to obtain a near optimal solution. Another main drawback from this method is it requires the number of clusters to be defined beforehand. If the data is not naturally clustered, strange results will be obtained.

Due to the drawbacks from the k-means algorithm, a new approach which is known as the global k-means algorithm has been proposed. This is a deterministic global optimization method that independent on any initial points by using k-means algorithms as a local search procedure [3].

Consider a clustering problem with a maximum cluster size k_{max}, the global k-means algorithm can proceed as below with a given data set $X=\{x_1, x_2,....,x_i\}$[3]:

1. K-means algorithm is performed on the data set started by setting the cluster size to be 1 (k=1) and slowly increase the cluster size until the maximum cluster size k_{max} is reached.
2. Starting from k=1, its optimal position is obtained which corresponds to the cluster center of the data set.
3. The algorithm proceeds by slowly increasing the number of cluster size k, where k< k_{max}. To solve the clustering problem for k, the k-1 centers are always placed at the optimal position that we obtain from the (k-1) clustering problem, which are $(m_1*(k\text{-}1),...,m_{k\text{-}1}*(k\text{-}1))$. For the kth center, i executions of k-means algorithm is performed with different initial positions stating from $X=\{x_1, x_2,....,x_i\}$ to obtain the best solution. By doing the above, the k clustering problem is solved by considering $(m_1*(k),...,m_k*(k))$ as the final solution.
4. The above steps will be repeated for other cluster size, starting from k=2 until $k_{max.}$

By using the above method, we can finally obtain a solution with k_{max} cluster and also solve the intermediate clustering problem, k=1, 2, 3...... k_{max}-1. Besides that, global k-means algorithm provides a good approach to discover the correct number of clusters by choosing the cluster size that gives the minimum clustering error. This solves the problem in the k-means algorithm which requires specification on the number of clusters before performing the algorithm.

3 Application of Dempster-Shafer Theory on Pattern Classifier

3.1 Dempster Shafer Theory

Evidence theory is based on belief functions and plausible reasoning, where belief measure is a form associated with preconceived notions and plausibility measure is a form associated with information that is possible [2]. These two parameters can be expressed and measured by using another function, known as Basic Belief Assignment (BBA).A basic belief assignment, denoted by m(A), represents the belief that a specific element x belongs to crisp set A, given a certain piece of evidence. Belief and plausibility measure can be expressed in terms of BBA by using:

$$bel(A) = \sum_{B \subseteq A} m(B) \tag{2}$$

$$pl(A) = \sum_{A \cap B \neq 0} m(B) \tag{3}$$

where bel(A) measures degree of belief that a given element x of universal set X belongs to the set A and to the various subsets of A and pl(A) represents not only the total belief that the element x belongs to A or to any of the subsets of A, but also the additional belief on the set that intersects with set A.

Separate pieces of evidence from two different sources (B and C) can be expressed by two different BBA, m_1 and m_2. These BBAs may be combined to obtain a new BBA by using Dempster-Shafer theory, which is defined as

$$m(0) = 0 \tag{4}$$

$$m(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{\sum_{B \cap C \neq 0} m_1(B) m_2(C)} \tag{5}$$

From Dempster-Shafer theory, a representation of the uncertainty of an element x is given, not only on a singleton but also on the union of the crisp sets.

3.2 Evidence Theoretic Classifier Based on Dempster-Shafer Theory

By referring to Thierry Denoeux work [4], a pattern classifier based on Dempster-Shafer theory has been presented. This classifier is composed of an input layer (L_1), two hidden layers (L_2 and L_3) and an output layer (L_4). Modification has been done on the classifier by using global k-means clustering algorithm as the clustering method. Consider a set of test patterns X={x_1, x_2,…,x_n} that need to be classified into one of the M classes, w_1,…,w_M where a set of known pattern training data is available for pattern classification task.

The first hidden layer L_2 contains of n prototype. Before classifier is tested using any benchmark data set, training data is needed for learning purpose. These classified training patterns are synthesized into a limited number of prototype p_1….p_n, by using global k-means algorithm. By applying this algorithm, the cluster center for each prototype can be discovered. The Euclidean distance between the testing pattern and the cluster center for each prototype is obtained using

$$d^i = \left\| x - p^i \right\| \tag{6}$$

Thus, the activation function of this layer is described by

$$s^i = \alpha^i \exp(-\gamma^i (d^i)) \tag{7}$$

This is a decreasing function corresponding to the distance between the test vector and each cluster (prototype) center.

The second hidden layer corresponds to the computation of BBA from n prototype. The distance between test patterns and each cluster center for the prototype can be regarded as a piece of evidence that influences the classification of the test pattern into one of the M classes, as each prototype will provide a degree of membership μ_q^i to each class. BBA can be treated as the belief for test vector to be categorized into one of the M classes. For a prototype p^i, the BBA which acts as the vector activation can be obtained by using

$$m^i = (m^i(\{w_1\}),....,m^i(\{w_M\}), m^i(\{\Omega\}) \tag{8}$$

$$\text{where} \quad \begin{aligned} m^i(\{w_q\}) &= \alpha^i \mu_q^i \phi^i(d^i) \\ m^i(\Omega) &= 1 - \alpha^i \phi^i(d^i) \end{aligned} \tag{9}$$

and Ω represents the frame of discernment which includes all the possible classes for the test vector x, $\phi^i(d^i)$ can be considered as a decreasing function of distance and μ_q^i is the degree of membership of the prototype toward certain class.

The n BBAs from each prototype are then combined at the output layer by using Dempster-Shafer rule of combination. The activation function μ^i in this layer is described by

$$\begin{aligned} \mu_j^i &= \mu_j^{i-1} m_j^i + \mu_j^{i-1} m_{M+1}^i + \mu_{M+1}^{i-1} m_j^i \\ \mu_{M+1}^i &= \mu_{M+1}^{i-1} m_{M+1}^i \end{aligned} \tag{10}$$

The BBA which contains the similar class from each prototype is combined to form μ^i.

Therefore the output vector can be defined as:

$$\text{m=}\mu^n \tag{11}$$

and the test pattern x can be assigned to the class which has the highest value of BBA:

$$m(\{w_i\}) = \max m(\{w_q\}) \tag{12}$$

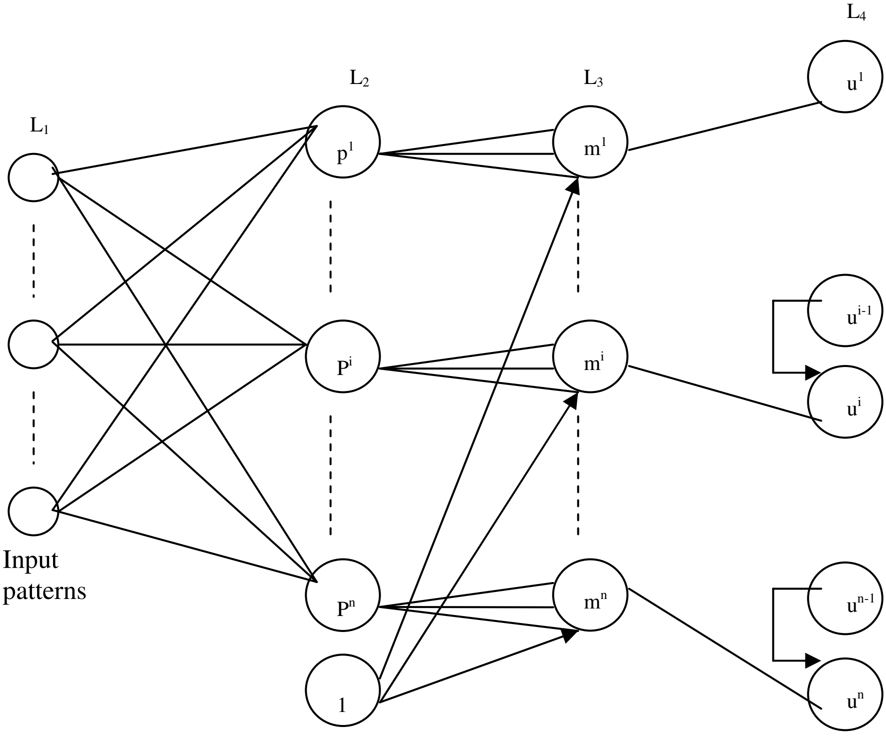


Fig. 1. Evidence theoretic classifier

As mentioned before, training set must be input into the classifier for learning purpose. The learning algorithm that has been chosen is the gradient decent method. This algorithm enables the convergence of the error function to a local minimum. The error function is defined by the difference between the classifier output and the target output value.

$$E_v\{x\} = \frac{1}{2}\left\|P_v - t\right\|^2 = \frac{1}{2}\sum_{q=1}^{M}(P_{v,q} - t_q) \tag{13}$$

where P_v is the classifier output and t is the target output vector.

4 Results and Discussion

Evidence theoretic classifier which used global k-means algorithm as the clustering approach is evaluated by using two data sets:

1. benchmark data which includes medical diagnosis data and iris data
2. Westland vibration data

Benchmark data

Benchmark data includes medical diagnosis data and iris data. Medical diagnosis data is obtained from [1]. The descriptions of the medical diagnosis data and iris data are as below:

Heart Disease data set

This data set contains 270 samples with 13 input features and 2 target classes, with 0 for absence of the disease and 1 for the presence of the disease.

Pima Indian Diabetes (PIMA) data set

This data set contain 2 target classes with class 1 indicated "tested positive for diabetes" and vice-versa.

Cancer data set

This data set contains 2 target classes.

Hepatobiliary Disorders(HEPATO) data set

This data consists of 68 patterns each with 7 input features, and 2 target classes, with 0 for absence of the disease and 1 for the presence of the disease.

Dermatology data set

The data set contains six target classes of Dermatology diseases (psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, cronic dermatitis, and pityriasis rubra pilaris).

Iris Data

The data set consists of three classes (iris virginica, iris versicolor, and iris setosa).

The common performance metrics used in medical diagnosis tasks are accuracy, sensitivity and specificity.

$$\begin{aligned}
&\text{Accuracy} = \frac{\text{Total number of correctly diagnosed cases}}{\text{Total number of cases}}\\
&\text{Sensitivity} = \frac{\text{Total number of positive cases correctly diagnosed}}{\text{Total number of positive cases}}\\
&\text{Specificity}=\frac{\text{Total number of negative cases correctly diagnosed}}{\text{Total number of negative cases}}
\end{aligned} \tag{14}$$

The positive case refers to the presence of a disease and the negative case refers to the absence of the disease. Accuracy is the measure of the ability of the classifier to provide correct diagnosis. Sensitivity measures the ability of the classifier to correctly identify the occurrence of a target class while specificity measures the ability to separate the target class.

Westland Vibration Data

The data set consists of vibration data collected using eight sensors mounted on different locations on the aft main power transmission of a US Navy CH-46E helicopter[13]. Data collected on torque level 100% from sensor 1 to sensor 4 is used to test the classifier.

The table below shows the performance of the classifier by using global k-means algorithms when compared to the k-means algorithm. The result of the k-means algorithm is obtained from 10 runs with different initial starting cluster centers.

Table 1. Data comparison between global k-means and k-means algorithm using benchmark data(medical diagnosis data and iris data)

		Accuracy			Sensitivity			Specificity		
		Mean	Max	Min	Mean	Max	Min	Mean	Max	Min
Heart	Global k-means	84.4	—	—	78.97	—	—	89.54	—	—
	k-means	84.16	84.4	84	78.36	78.90	78.13	89.62	90.31	89.54
Cancer	Global k-means	96.54	—	—	93.51	—	—	98.24	—	—
	k-means	96.32	96.35	96.15	93.01	93.06	92.64	98.24	98.24	98.24
Pima	Global k-means	78	—	—	58.62	—	—	87.66	—	—
	k-means	77.6	77.86	77	58.73	59.49	57.82	86.98	87.89	85.93
Hepato	Global k-means	48.98	—	—	—	—	—	—	—	—
	k-means	46.67	48.16	45.51	—	—	—	—	—	—
Dermatology	Global k-means	61.3	—	—	—	—	—	—	—	—
	k-means	57.65	61.91	45.95	—	—	—	—	—	
Iris	Global k-means	57.86	—	—	—	—	—	—	—	—
	k-means	55.36	57.14	47.14	—	—	—	—	—	—

Table 2. Data comparison between global k-means and k-means algorithm using Westland Vibration data(sensor 1 to sensor 4)

Westland Vibration data				
		Accuracy		
		Mean	Max	Min
Sensor 1	Global k-means	98.59	––	––
	k-means	80.52	86.62	73.67
Sensor 2	Global k-means	96.62	––	––
	k-means	81.70	89.58	75.21
Sensor 3	Global k-means	97.61	––	––
	k-means	87.65	93.66	79.72
Sensor 4	Global k-means	96.06	––	––
	k-means	74.66	78.59	70.56

The result shows that the global k-means algorithm provides higher accuracy when compare with k-means algorithm. The reason for using global k-means algorithm is that it will overcome the shortcoming in k-means algorithm:

1. The initial starting point of k-means algorithm is selected randomly. Thus it does not guarantee unique clustering and where each run will give different result depends on the initial position. Due to this, it does not yield a convergence to a global optimum. This problem can be solved by using global k-means algorithm which is not sensitive to any initial starting condition.
2. The k-means algorithm needs the user to specify the number of clusters. While global k-means algorithm can be used to discover the optimized cluster size by choosing the cluster size that gives the minimum clustering error criterion.

5 Conclusion

An evidence theoretic classifier has been presented by using global k-means algorithm as the clustering method. The results show that this classifier gives a comparable and good performance when compared to k-means clustering algorithm. Global k-means algorithm overcomes some problem that one could face when applying k-means algorithm, especially the randomness of the initial cluster center. Due to the randomness in k-means algorithm, one is forced to run the algorithm for several times in order to obtain a near optimal result. Moreover, global k-means algorithm enables the user to discover the actual number of cluster sizes. Therefore this method is considered as another effective alternative for cluster initialization for evidence-based neural network.

References

1. Chu Kiong Loo, M.V.C. Rao.: Accurate and reliable diagnosis and classification using probabilistic ensemble Simplified Fuzzy Artmap. IEEE Transactions on Knowledge and Data Engineering. Vol.17. NO.11 (2005)
2. Timothy J.Ross.: Fuzzy Logic with Engineering Applications. John Wiley & Sons, Ltd
3. Aristidis Likas, Nikos Vlassis, Jakob J.Verbeek.: The global k-means clustering algorithm. Pattern recognition. **36** (2003) 451-461
4. Thierry Denoeux.: A neural network classifier based on Dempster-Shafer theory. IEEE Transaction on systems. MAN and Cbernetics-Part A: Dydtem and Humans. Vol 30. No.2 (2000)
5. Lalla Meriem Zouhal, Thierry Denoeux.: An evidence-theoretic k-NN rule with parameter optimization. IEEE Transaction on systems, MAN and Cybernetics-Part C: Applications and Reviews. Vol 28. No.2. (1998)
6. Shafer, G.: A mathematical theory of evidence. Princeton. NJ: Priceton, Univ.Press (1976)
7. Denoeux, T.: A k-nearest neighbor classification rule based on Dempster-Shafer theory. IEEE Trans. Syst., Man, Cybern. Vol25, (1995) 804-813
8. J.A. Lozano, J.M. Pena, P.Larranaga: An empirical comparison of four initialization methods for the K-means algorithm". Pattern Recognition letters 20(2002), 77-87
9. Shehroz S. Khan, Amir Ahmad: Cluster center initialization algorithm for K-means clustering. Pattern recognition letters. **25** (2004) 1293-1302
10. Huadong Wu, Mel Diegel, Rainer Stiefelhagen, Jie Yang.: Sensor fusion using Dempster-Shafer theory. IEEE Instrumentation and Measurement Technology Conference. (2002)

11. S.Le Hegarat-Mascle, Richard, D., Ottle, C.: Multi-scale data fusion using Dempster Shafer Evidence theory. Integrated Computer-Aided Engineering 10(2003) 9-22
12. J. B. MacQueen : Some methods for classification and analysis of multi variate Observations. Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley. University of California Press. (1967) 1:281-297
13. Gary G.Yen, Phayung Meesad.: An effective neuro-fuzzy paradigm for machinery condition health monitoring. IEEE Transactions on systems, MAN and Cybernetics-Part B: Cybernetics. Vol.31. No.4. (2001)

Bark Classification Based on Gabor Filter Features Using RBPNN Neural Network

Zhi-Kai Huang[1,2], De-Shuang Huang[1], Ji-Xiang Du[1,2],
Zhong-Hua Quan[1,2], and Shen-Bo Guo[1,2]

[1] Intelligent Computing Lab, Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, P.O. Box 1130, Hefei, Anhui 230031, China
[2] Department of Automation, University of Science and Technology of China
huangzk@iim.ac.cn

Abstract. This paper proposed a new method of extracting texture features based on Gabor wavelet. In addition, the application of these features for bark classification applying radial basis probabilistic network (RBPNN) has been introduced. In this method, the bark texture feature is firstly extracted by filtering the image with different orientations and scales filters, then the mean and standard deviation of the image output are computed, the image which have been filtered in the frequency domain. Finally, the obtained Gabor feature vectors are fed up into RBPNN for classification. Experimental results show that, first, features extracted using the proposed approach can be used for bark texture classification. Second, compared with radial basis function neural network (RBFNN), the RBPNN achieves higher recognition rate and better classification efficiency when the feature vectors have low-dimensions.

1 Introduction

Plant species identification is a process resulting in the assignment of each individual plant to a descending series of groups of related plants, as judged by common characteristics. It is important and essential to correctly and quickly recognize and identify the plant species in collecting and preserving genetic resources, discovery of new species, plant resource surveys and plant species database management, etc. Plant identification has had a very long history, from the dawn of human existence. However, so far, this time-consuming and troublesome task was mainly carried out by botanists. Currently, automatic plant recognition from color images is one of the most difficult tasks in computer vision because of lacking of proper models or representations for plant. In addition, different plants take on numerous biological variations, which farther increased the difficult of recognition.

Many plant barks show evident texture features, which can be used as one of useful features for plant recognition. From bark texture analysis, we can conclude that it is necessary to define a set meaningful feature for exploring the characteristics of the texture of bark. There have been several approaches for this problem such as spatial gray-level co-occurrence matrix [1], Gabor filter banks [2], combining grayscale and

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 80 – 87, 2006.

binary texture [3]. Although these methods yield a promising result to bark texture analysis, but they fail to classify bark texture adequately. One of the most popular signal processing approaches for texture feature extraction is Gabor filters which can filtering both in the frequency and spatial domain. It has been proposed that Gabor filters can be used to model the responses of the human visual system. A bank of filters at different scales and orientations allows multichannel filtering of an image to extract frequency and orientation information. This can then be used to decompose the image into texture features.

2 Gabor Wavelets and Feature Extraction

2.1 Gabor Wavelets

A 2-D Gabor function is a Gaussian modulated by a complex sinusoid [4]. It can be specified by the frequency of the sinusoid ω and the standard deviation σ_x and σ_y, of the Gaussian envelope as:

$$g(X,Y)=\frac{1}{2\pi\sigma_x\sigma_y}\cdot\exp[-\frac{1}{2}(\frac{X^2}{\sigma_x^2}+\frac{Y^2}{\sigma_y^2})+2\pi j\omega X] \tag{1}$$

The frequency response of this filter is written as:

$$G(U,V)=\exp\{-\frac{1}{2}[\frac{(U-\omega)^2}{\sigma_u^2}+\frac{V^2}{\sigma_v^2}]\} \tag{2}$$

Where $\sigma_u=\frac{1}{2\pi\sigma_x},\sigma_u=\frac{1}{2\pi\sigma_x}$

The self-similar Gabor wavelets are obtained through the generating functions:

$$\begin{gathered}g_{mn}(X,Y)=a^{-m}\cdot g(X',Y')\\ X'=a^{-m}(X\cos\theta+Y\sin\theta),Y'=a^{-m}(-X\sin\theta+Y\cos\theta)\end{gathered} \tag{3}$$

$$\theta=\frac{n\pi}{N},a>1,m,n=Intergers$$

Where m and n specify the scale and orientation of the wavelet, respectively, with $m=0,1,2,...M,n=0,1,2,...N-1$ and M,N are the total number of scales and orientations.

The Gabor kernels in Eq.1 are all mutually similar since they can be generated from the same filter, also known as mother wavelet. As described above, Gabor filter can localize direction spatial frequency at θ. When applied to an image, the output responds maximally at those particular edges whose orientation is θ. That means Gabor filter is oriental selective to image. We can use this specialty to detect the edges at all orientations of an image.

2.2 Image Feature Extraction

The Gabor wavelet image representation is a convolution of that image within the same family of Gabor kernels in Eq.1. Let $I(x,y)$ be the gray level distribution of an image, and the convolution of image I together with a Gabor kernel g_{mn} is defined as follows:

$$W_{mn}(x,y) = \int I(x,y) g^{*}_{mn}(x-x_1, y-y_1) dx_1 dy_1 \tag{4}$$

Where * indicates the complex conjugate and W_{mn} is the convolution result corresponding to the Gabor kernel at orientation m and n . It is assumed that the local texture regions are spatially homogeneous, and the mean μ_{mn} and the standard deviation σ_{mn} of the magnitude of the transform coefficients are used to represent the region for classification purposes: $\mu_{mn} = \int\int |W_{mn}(x,y)| dxdy$ and $\sigma_{mn} = \sqrt{\int\int (|W_{mn}(x,y)| - \mu_{mn})^2 dxdy}$

A feature vector is now constructed using the mean μ_{mn} and standard deviation σ_{mn} of the output in the frequency domain as feature components.

$$\bar{f} = [\mu_{00}\sigma_{00}\mu_{01} \cdots \mu_{35}\sigma_{35} \cdots] \tag{5}$$

We use this feature vector as bark recognition feature vector.

3 Radial Basis Probabilistic Network (RBPNN) Classifier

After the Gabor features of bark image have been extracted which had described in section 2, the second task is that recognition of bark texture image using radial basis probabilistic network (RBPNN).

The RBPNN model is essentially developed from the radial basis function neural networks (RBFNN) and the probabilistic neural networks (PNN) [5], [6], [7], [8]. Therefore, the RBPNN possesses the common characteristics of the two original networks, i.e., the signal is concurrently feed-forwarded from the input layer to the output layer without any feedback connections within the network models. Moreover, the RBPNN avoids the disadvantages of the two original models to some extent. The RBPNN, shown in Fig.1, consists of four layers: one input layer, two hidden layers and one output layer. The first hidden layer is a nonlinear processing layer, which generally consists of hidden centers selected from a set of training samples. The second hidden layer selectively sums the first hidden layer outputs according to the categories to which the hidden centers belong. Generally, the corresponding weight values of the second hidden layer are 1's. For pattern recognition problems, the outputs in the second hidden layer need to be normalized. The last layer for RBPNN is simply the output layer, which completes the nonlinear mapping by carrying out

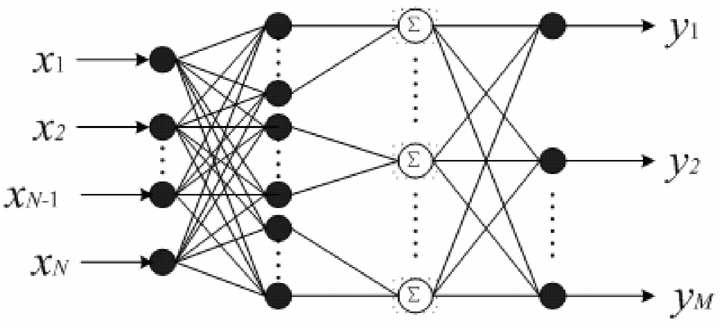


Fig. 1. The topology scheme of the RBPNN

tasks such as classification, approximation and prediction. In fact, the first hidden layer of the RBPNN has the vital role of performing the problem-solving task.

Training of the network for the RBPNN used orthogonal least square algorithms (OLSA). The advantages of recursive least square algorithms are that it can fast convergence and good convergent accuracy. The algorithms can be expressed as the following equation in mathematics:

$$y_i^o = \sum_{k=1}^{M} w_{ik} h_k(x) \tag{6}$$

$$h_k(x) = \sum_{i=1}^{n_k} \phi_i(x, c_{ki}) = \sum_{i=1}^{n_k} \phi_i(\|x - c_{ki}\|_2) \quad , k = 1,2,\cdots M \tag{7}$$

Here, x is a given input vector, y_i^o is the output value of the i -th output neuron of neural network, $h_k(x)$ is the k -th output value of the second hidden layer of network; w_{ik} is the weight matrix between the k -th neuron of the second hidden layer and the i -th neuron of the output layer, c_{ki} represents the i -th hidden center vector for the k -th pattern class of the first hidden layer; n_k represents the number of hidden center vector for the k -th pattern class of the first hidden layer; $\|\bullet\|_2$ is Euclidean norm; and M denotes the number of the neurons of the output layer and the second hidden layer, or the pattern class number for the training samples set; $\phi_i(\|x - c_{ki}\|_2)$ is the kernel function, which is generally Gaussian kernel function can be written as.

$$\phi_i(\|x - c_{ki}\|_2) = \exp(-\frac{\|x - c_{ki}\|_2^2}{\sigma_i^2}) \tag{8}$$

For m training samples, Eq.6 can be expressed as:

$$\begin{bmatrix} y_{11}^{o} & y_{12}^{o} & \cdots & y_{1m}^{o} \\ y_{21}^{o} & y_{22}^{o} & \cdots & y_{2m}^{o} \\ \cdots & & & \\ y_{n1}^{o} & y_{n2}^{o} & & y_{nm}^{o} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1m} \\ h_{21} & h_{22} & \cdots & h_{2m} \\ \cdots & & & \\ h_{n1} & h_{n2} & & h_{nm} \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \cdots & & & \\ w_{m1} & w_{m2} & & w_{mm} \end{bmatrix} \tag{9}$$

that also can be writed as:

$$Y^{O} = HW \tag{10}$$

From [7], the weight matrix W can be solved as follows:

$$W = R^{-1}\hat{Y} \tag{11}$$

where $R, \overset{\frown}{Y}$ can be obtained as follows:

$$H = Q\begin{bmatrix} R \\ L \\ 0 \end{bmatrix}, Q^{T}Y = \begin{bmatrix} \overset{\frown}{Y} \\ \tilde{Y} \end{bmatrix} \tag{12}$$

where Q is an $n \times n$ orthogonal matrix with orthogonal columns satisfying $QQ^{T} = Q^{T}Q = I$, and R is an $m \times m$ upper triangle matrix with the same rank as H . In Eq. (11), $\overset{\frown}{Y}$ is a $(N-M)\times M$ matrix. Equation (11) expresses the orthogonal decomposition of the output matrix H of the second hidden layer of RBPNN.

4 Image Data and Experimental Results

4.1 Image Data and Features Chosen

We have collected more than 300 pictures of bark in our image database. These images were recorded at a resolution of 640 x 480 pixels, with a bit depth of 16 bits/pixel. Thus, 256 levels were available for each R, G, and B color plane. The images were converted to JPEG format and grayscale intensity image before processing. Some bark images are shown in Fig.2.

Chosen randomly about 50% of plant bark samples for each bark class form a testing set and the remaining samples form a training set. By this partition, there are 248 samples in the training set and 17 character samples in the testing set. In addition, because the trunk of the tree is cylinder and the two sides of the pictures are possibly blurred, so the particularity of interests (ROI), we have select that is a relatively bigger ROI with the size of 350×400 pixels.

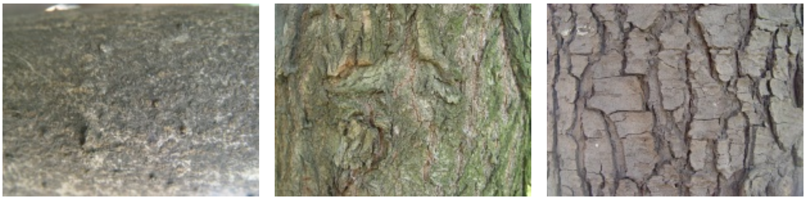


Fig. 2. Three kinds of original bark images

As we have discussed in section 2, the Gabor filter-based feature extraction method requires setting control parameters of Gabor filter. Hence a feature vector consists of different parameters will be obtained which contains the visual content of the image. To get the best result, the Gabor parameters were test for different values of the number of scales (m) and the number of orientations (n). The average recognition rates have been presented in Table 1.

The experiment has been made on a PC (PentiumIV-2.4GHz CPU, 512M RAM).The image features were calculated using subroutines written in Matlab 7.0 language. Software for Classifier of RBPNN, we use a conventional C++6.0 programming environment. Totally seventeen bark classes are used for identification. These were: retinispora, maple, Sophora japonica, dogbane, trumpet creeper, osier, pine, phoenix tree, camphor, poplar and willow, honey locust, palm, gingkgo, elm, etc. Every type of bark has half images for training, others for testing. We used the" quantity average recognition rate" defined as below to compare the results.

$$\text{Average Recognition Rate} = \frac{\text{Number of Bark Image Classified Truely}}{\text{Totat Number of Classified Bark Images}} \cdot \%$$

The obtained average recognition rates are presented in Table 1.

Table 1. Average Recognition Rates for Different Gabor Filter and SVM classifier

Gabor Filter Features Used	RBPNN	SVM
Orientation(n)=6, Scales(m)=4	63.71%	60.48%
Orientation(n)=6, Scales(m)=5	72.58%	78.22%
Orientation(n)=6, Scales(m)=6	79.03%	81.45%
Orientation(n)=6, Scales(m)=7	77.42%	83.06%
Orientation(n)=5, Scales(m)=4	62.90%	62.10%
Orientation(n)=5, Scales(m)=5	74.19%	77.42%
Orientation(n)=5, Scales(m)=6	76.61%	79.84%
Orientation(n)=5, Scales(m)=7	79.84%	81.45%
Orientation(n)=4, Scales(m)=4	66.13%	64.52%
Orientation(n)=4, Scales(m)=5	77.42%	76.61%
Orientation(n)=4, Scales(m)=6	79.84%	80.64%
Orientation(n)=4, Scales(m)=7	79.84%	82.26%
Orientation(n)=4, Scales(m)=8	80.65%	81.45%

From the classification performances shown in Table 1, we found that: 1) for each fixed spatial sampling resolution, there exists an optimal wavelength which achieves the best performance. We observed that the orientation $n = 4$ achieves the better performance in bark recognition experiments. 2) While orientation $n = 4$ and scales increasing, the average recognition rate of bark can be improved. 3) As for the spatial sampling resolutions, it seems that 4×8 sampling is enough for bark classification when used RBPNN classifier. Adopting the more Gabor image feature that can improve accuracy of bark classification, but it will lead to a time-consuming computation.4)In order to compare the effectiveness of the Gabor features with that of the other classifier such as SVM, Our results show that RBPNN classifier can achieve more better when the feature vectors has low-dimensions such as the dimensions is fewer than 24,at the same time the SVM classifier can give better classification accuracy while the dimension of feature vectors is above 24.

5 Conclusion

This paper proposes a bark texture recognition algorithm, in which Gabor feature representation and RBPNN classifier are employed. The neural network which was trained using orthogonal least square algorithms is employed to classify such feature vectors and tested on different scales and different orientation. We have also found in experiments that RBPNN offers an accuracy of higher classification when the feature vectors have low-dimensions such as the dimensions is fewer than 24. When the dimension of the feature vectors is high, the RBPNN can give similar results as SVM. In the future, more effective feature extracted methods will be investigated for bark classification.

References

1. David, A. Clausi, Huang, D..: Design-Based Texture Feature Fusion Using Gabor Filters and Co-Occurrence Probabilities, IEEE Transactions on Image Processing, Vol. 14, No. 7, July (2005) 925-936
2. Zheru Chi, Li H. Q., Wang C..: Plant Species Recognition Based on Bark Patterns Using Novel Gabor Filter Banks IEEE Int. Conf. Neural Networks & Signal Processing Nanjing. China, (2003) 1035-1038
3. Wan, Y. Y., Du, J. X., Huang, D. S., Chi, Z. R.: Bark Texture Feature Extraction Based on Statistical Texture Analysis, International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong, (2004) 482-185
4. Manjunath, B. S., Ma, W.Y.: Texture features for Browsing and Retrieval of Image Data IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI - Special issue on Digital Libraries), vol. 18, no. 8, Aug(1996) 837-842
5. Huang, D.S.: Radial Basis Probabilistic Neural Networks: Model and Application, International Journal of Pattern Recognition and Artificial Intelligence, 13(7), (1999) 1083-1101
6. Gamm, J. B, D. L. Yu.: Selecting Radial Basis Function Network Centers with Recursive Orthogonal Least Squares Training, IEEE Trans. Neural Network. vol. 11, No. 2, Mar. 2000

7. Zhao, W.B, Huang, D.S.: Application of Recursive Orthogonal Least Squares Algorithm to the Structure Optimization of Radial Basis Probabilistic Neural Networks, The 6th International Conference on Signal Processing (ICSP02), Beijing, China, (2002) 1211-1214
8. Huang, D. S.: Systematic Theory of Neural Networks for Pattern Recognition. Publishing House of Electronic Industry of China, Beijing, (1996)

A Hybrid Handwritten Chinese Address Recognition Approach

Kaizhu Huang[1], Jun Sun[1],
Yoshinobu Hotta[2], Katsuhito Fujimoto[2], Satoshi Naoi[2],
Chong Long[3], Li Zhuang[3], and Xiaoyan Zhu[3]

[1] Information Technology Lab, Fujitsu R&D Center Ltd, Beijing, China
{kzhuang, sunjun}@cn.fujitsu.com
[2] Fujitsu Laboratories Ltd, Kawasaki, Japan
{y.hotta, fujimoto.kat, naoi.Satoshi}@jp.fujitsu.com
[3] Dept. of Computer Science and Technology, Tsinghua University, Beijing, China
{longc05, zhuangli98, zxy-dcs}@mails.tsinghua.edu.cn

Abstract. Handwritten Chinese Address Recognition describes a difficult yet important pattern recognition task. There are three difficulties in this problem: (1) Handwritten address is often of free styles and of high variations, resulting in inevitable segmentation errors. (2) The number of Chinese characters is large, leading low recognition rate for single Chinese characters. (3) Chinese address is usually irregular, i.e., different persons may write the same address in different formats. In this paper, we propose a comprehensive and hybrid approach for solving all these three difficulties. Aiming to solve (1) and (2), we adopt an enhanced holistic scheme to recognize the whole image of words (defined as a place name) instead of that of single characters. This facilitates the usage of address knowledge and avoids the difficult single character segmentation problem as well. In order to attack (3), we propose a hybrid approach that combines the word-based language model and the holistic word matching scheme. Therefore, it can deal with various irregular address. We provide theoretical justifications, outline the detailed steps, and perform a series of experiments. The experimental results on various real address demonstrate the advantages of our novel approach.

1 Introduction

We consider the problem of Handwritten Chinese Address Recognition (HCAR). HCAR describes a difficult yet important problem in pattern recognition. This technology can be applied in many domains including postal address recognition and bank cheque recognition. The basic task of this problem is to recognize the actual address directly from an input address image, which can be obtained by either a scanner or a digital camera.

Many methods have been proposed for dealing with this problem. Most of them are based on the so-called plain recognition approach [2][4]. This method first segments each character from the input image, and then recognizes the

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 88–98, 2006.

isolated characters one by one. However, there are many difficulties for this approach. For example, free styles in handwriting will lead to inevitable segmentation errors; the large number of categories in Chinese characters[1] will cause high misclassification rate for isolated character recognition. Moreover, this bottom-up recognition strategy (e.g., from single characters recognition to the address) makes it difficult to utilize the address knowledge.

Aiming to solve the problems caused by plain recognition, some researchers have applied the holistic word recognition approach [5][7]. This approach firstly extracts the key characters, which are defined as the basic administration units, such as 省 (province), 市 (city), 区 (district), and 路 (road). It then bases the address knowledge to holistically recognize the words, which are defined as a sequence of characters (place names) between two key characters. Figure 1 illustrates this approach. The key characters 市 (city), 区 (district), and 路 (road) are firstly extracted from the image. Then the word images between each pair of key characters or before the first key character (the image of the word 上海) are segmented. These images are then holistically matched with the synthesized features of the place names (stored in the reference dictionary). After 上海 is recognized, the word image of 松江 is extracted and matched with the synthesize features of all the place names of 区 which are located in 上海. The process is repeated until all the words are recognized. Finally, the real string 上海-市-松江-区-松乐-路 is output. One problem of this method is that, to reduce the time complexity, only the first candidate of the word recognition is adopted. This candidate will serve as the upper address for recognizing the place name in the next address level. If a word is mis-recognized in one level, the word recognition will be definitely incorrect in the next level. This is because the candidate of word recognition is only searched within the place specified by the previous level. For example, in Fig 1, the second level word image (i.e., the word image of

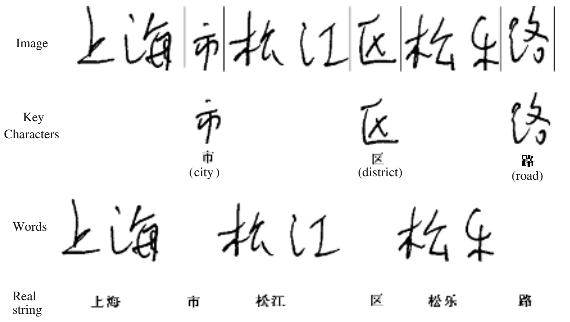


Fig. 1. A typical example of key characters and words

[1] Typically, there are 6763 categories for the first and second level simplified Chinese characters.

"松江") will be matched with all the "区 (district)" that are located in "上海" (as given by the first candidate in the first level). If the first candidate is not "上海", the recognition would fail immediately. Another even more critical problem is that this approach cannot deal with irregular address, namely those address with missing key characters. Some users often write address strings without key characters. In this event, this holistic method would absolutely fail. Fig. 2 describes two strings that represent the same address. (a) is a regular address; (b) is an irregular address with missing key characters 市 and 区. Both address strings are commonly seen in real cases.

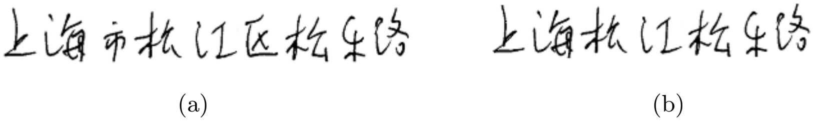


Fig. 2. An illustration of regular address and irregular address. Two strings describe the same address.

As a brief summary in the above, there are mainly three difficulties in HCAR. (1) Handwritten address is often of free styles and of high variations, resulting in inevitable segmentation errors. (2) The number of Chinese characters is large, leading low recognition rate for single Chinese characters. (3) Chinese address is usually irregular, i.e., different persons may write the same address in different formats.

Aiming to solve all the three above difficulties, we propose a novel hybrid method. We first design an enhanced holistical word matching approach. This approach not only overcomes the difficulties of (1) and (2), but more importantly, based on exploiting a recursive scheme and a verification technique, it also solves the problem caused by only using the first candidate in word matching [5][7], therefore providing the potentials to increase the recognition rate significantly. To solve the problem of (3), we also develop a Word Based Language Model (WBLM) that is tailored to deal with irregular address. WBLM calculates the probability that a word occurs after another word. The advantage is that it searches the words based on the probability and does not need key characters to set the word boundary. Finally, these two approaches are seamlessly integrated such that the whole system can deal with all the three difficulties, thus representing a comprehensive approach for HCAR.

This paper is organized as follows. In Section 2, we present the enhanced holistic word matching approach. In Section 3, we explain the Word Based Language Model in details. In Section 4, we present how to combine these two approaches in order to overcome all the three difficulties. In Section 5, we then evaluate our proposed method against other traditional methods on real address. Finally, we set out the conclusion.

2 Enhanced Holistic Word Matching Approach

In this section, we first make a simple review on the traditional holistic word matching approach. We then in Section 2.2 present the new recursive approach. Next, we discuss a speed-up strategy in Section 2.3. Following that, a verification method is proposed to further lift the system performance.

2.1 Review of Traditional Holistic Word Matching

First, key characters will be extracted from the address image. In Chinese address, there are only 22 key characters which are 市, 省, 区, 弄, 路, 街, 村, 乡, 镇, 港, 湾, 县, 道, 里, 同, 巷, 楼, 州, 旗, 胡, 庄, 坊. Details about how to extract the key characters can be seen in [7]; After the key character is extracted, the words (place name) between each pair of key characters will be holistically recognized. Different from the plain address recognition method, this traditional approach recognizes the images between each pair of key characters as a whole. Beginning from the first address level, the word image is segmented and recognized as a place name. In the next address level, the word image is cut out and the features are extracted from it. These features are then compared with the synthesized features of those place names; these place names must be those of the administrative units specified by the key character in this level and must locate in the place specified by the recognition result in the previous level. Similar process is conducted until all the address levels are recognized. This scheme avoids the difficult single character segmentation problem and therefore increases the accuracy of the system. Detailed information about the feature synthesis and holistic matching can be seen in [5].

2.2 Recursive Holistic Word Matching

As seen in [6][5][7], the above holistic word recognition only adopts the first candidate in each level. However, handwriting is of free styles. Moreover, sometimes, two words (place names) contain very similar shapes. Only choosing the first candidate may generate many errors. Therefore, we propose an enhanced holistic word recognition approach. Our approach utilizes multiple candidates and recursively performs holistic word recognition in each address level. For solving the speed problem, we propose a trimming-down strategy, which will be introduced in Section 2.3.

Fig. 3 illustrates the detailed recursive procedure. In this figure, assuming the key characters 市 , 区 , and 路 be extracted, the words are W_1, W_2, and W_3. Firstly, features are extracted from the image of W_1 and then they are matched with all the 市 as indicated by A (e.g., 北京, 上海) in the reference dictionary. The candidate words are 上海，上饶，and 北海, which are sorted by matching distances. Each candidate will be recursively evaluated in matching the image W_2. For example, the image of W_2 is compared with all the place names which are 区 (as indicated by B) and are located in 上海市, 上饶市, 北海市 respectively. As a result, three candidate lists of W_2 are generated for 上海市, 上饶市, and 北海市 respectively. Similar process will be conducted on these

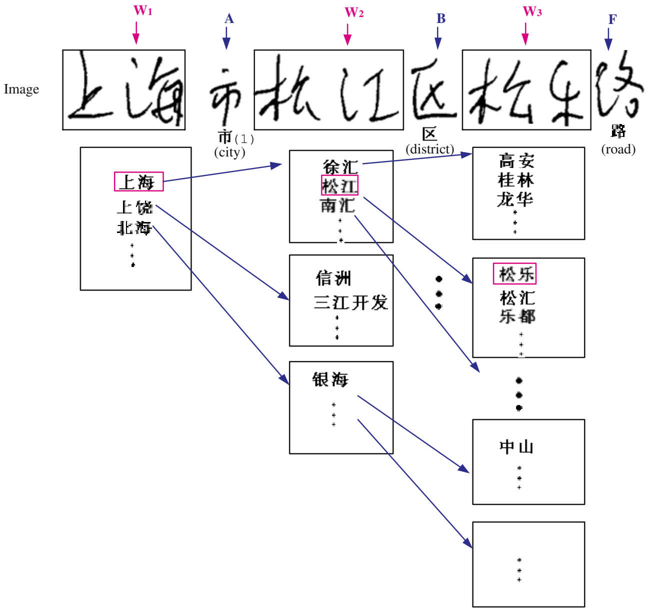


Fig. 3. Recursive and holistic word recognition

candidates to match the image of W$_3$. The recognition result is the path which has the smallest matching distance. In this example, it is 上海-市-松江-区-松乐-路.

Note that, the first recognition candidate of W$_2$ is not the correct word recognition result 松江. By using the recursive matching, it is selected as the recognition result. In such a way, the recognition accuracy will be lifted.

2.3 Trimming

In the above, we adopt the multiple-candidate strategy to increase the system's accuracy. However, the time complexity will be increased simultaneously. Assume there are k levels in an input address and N candidates are used in recognizing each word address. The total number of combinations is hence N^k. This will be very time-consuming. To speed up the whole process, we design a trimming-down strategy as follows:

Rule 1: The maximum number of candidates should be smaller than a given number K.
Rule 2: Only the candidates that satisfy the condition $\frac{Dist(Cand_i)-Dist(Cand_1)}{Dist(Cand_1)} < Th_{tr}$ will be evaluated.

In the above, $Dist(Cand_i)$ represents the matching distance of the i-th candidate. Th_{tr} is a predefined threshold, which is set to 0.125 in our system. Rule 1 will restrict the maximum number of the candidates. Rule 2 will not search those candidates whose matching distances are far from the first candidate. In this way, not all the combinations will be evaluated. Hence, the processing time can be reduced greatly.

2.4 Verification

When the words in the upper levels are mis-recognized, the word images in the latter levels will be absolutely mis-recognized. Hence the matching distances in the latter levels will be large. This means the recognition result of the latter levels can verify the recognition of upper levels. However, for the last level word recognition, there is no such verification. Moreover, as counted in practice, recognition errors occurring in the last level address accounts for 25% of misclassification. Therefore, for correcting the last level address, we propose to combine the plain character recognition result with the holistic word recognition result. We first provide a definition of the modified edit distance, which is different from its traditional definition.

Definition 1. ***Modified Edit Distance.*** *Assume that $S = \{S_1, S_2, \ldots, S_u\}$ is a place name, and W is a $C \times K$ array, where S_k $(1 \leq k \leq u)$ represents the k-th character in the place name, W_{ij} $(\,1 \leq i \leq C, 1 \leq j \leq K)$ represents the j-th candidate for the i-th segmentation part in a word image, W_i represents the candidate list of the single character recognition result for the i-th segmentation part. The modified edit distance between S and W is defined as the minimum cost at which W is changed to S by the operations of insertion, substitution, and deletion. The cost between S_k and W_i is defined as follows:*

$$Cost(S_k, W_i) = \begin{cases} 1 & if\ \forall j \quad W_{ij} \neq S_k \\ \frac{j}{CK} & if\ \exists j \quad W_{ij} = S_k \end{cases}. \tag{1}$$

In the above, C can be considered as the total number of the segmentation parts for the word images in the last level, and K can be regarded as the maximum number of candidates for each segmentation part.

In this procedure, in order to utilize the candidates of the plain results for the last level address image, we modify the concept of the edit distance which is designed originally for comparing two strings to that for comparing a string (a place name or address) and the string array (the plain recognition result and its candidates).

The detailed verification steps are described in the following:

1. Base the Dynamic Programming to calculate the modified edit distance between each place name in the last level and the plain recognition result for the last level image. All these place names are sorted according to their modified edit distances.
2. Get the word recognition results from fine recognition and rank each place name according to the matching distance.
3. Output the legal address A_f as the final recognition result according to the following decision rule as defined in Definition 2.

Definition 2. ***Decision Rule.*** *(1) If a place name contains the edit distance smaller than or equal to* 1*, output the place name with the minimum modified edit distance as the recognition result. (2) If all the place names contain the edit distance greater than or equal to* C*, the number of the connected components, output the word recognition result as the final result. (3) If (1) and (2) are not satisfied, the weight is calculated as Eq. (2) and output the place name with the smallest weight as the final result.*

$$Weight(A_i) = (1 - t_1)Rank_{ED}(A_i) + t_1 Rank_{WR}(A_i) \tag{2}$$

where, A_i is i-th place name, $Rank_{ED}(A_i)$ means the rank of the modified edit distance between A_i and the plain recognition result W, $Rank_{WR}(A_i)$ means the rank of A_i by the word recognition. t_1 is defined as $t_1 = round(ed(A_i))/C$; $ed(A_i)$ represents the modified edit distance between A_i and the plain recognition result.

The decision rule is justified in the following. From the cost definition in Eq. (1), with Rule (1) satisfied for a place name, each character in the legal address should occur in the candidate list of plain recognition. This actually implies a verification for this place name. Therefore we should output the place name with the minimum modified edit distance as the recognition result. On the other hand, with Rule (2) satisfied for all place names, none of the characters in these place names occurs in the candidate list of the plain recognition result. This actually implies that the plain recognition is highly unreliable. Therefore, the holistic word recognition result should be output. Rule (3) actually combines the plain recognition result with the holistical word recognition result. When the edit distance of A_i is very small, the plain recognition result appears highly reliable. Therefore we should give more weight to $Rank_{ED}(A_i)$; otherwise, $Rank_{WR}(A_i)$ should make more contributions. In particular, when the edit distance of A_i is less than or equal to 1, we should trust the result of plain recognition; when the edit distance of all the legal address is big enough, the plain recognition should be very unreliable, leading that we should output the word recognition result as the final result.

3 Word Based Language Model Approach

Statistical language models (SLM) receive much interest as the speed and capability of computers increases dramatically [3]. SLM can mine the inner rules by statistically and automatically analyzing a large quantity of data, called corpus. Traditional language models are usually based on analyzing the relationship among the characters. However, this type of character based models may be less effective in address recognition. In Chinese address, the basic meaningful units are usually words instead of characters. For example, in an address string “湖北(省)黄石市广场路” (the key character province 省 is missing), the basic units are “湖北”, “黄石市”, and “广场路”, which respectively represent three place names. Moreover, the major relationship that 黄石市 is located in 湖北(省) is described by words. If the relationship of single characters is considered, the probability that 山 appears after 黄, i.e., 黄山, (a famous mountain) is bigger than 黄石. However, 黄山 is not located in 湖北省. This may hence result in errors. In this paper, the word based language model is hence adopted.

3.1 Graph Search Algorithm Based on WBLM

When an address image is input, connected components (CC) will be firstly extracted based on Connected Component Analysis [1]. Let the component sequence be $\{C_1, C_2, \ldots, C_k\}$. For each sequence position i, the component C_i, the segment by combining C_i with C_{i+1} (if $i+1 \leq k$), and the segment by combining C_i, C_{i+1}, and C_{i+2} (if $i+2 \leq k$) will be input to the recognizer for classification. Therefore, the connected component sequence will form a Markov graph with each CC as the node. Moreover, each node will be decided only by its previous two nodes. Then a word graph will be constructed from the Markov graph by using the address knowledge tree as plotted in Fig. 4. In order to overcome the difficulty of missing key characters, a place name with and without the associate key characters are both embedded in the tree. Now the task changes into finding the optimal path with the maximum probability in the word graph. In the following, we describe how to calculate the probability of a certain path.

First, the probability that a q-length segment string $\{s_1, s_2, \ldots, s_q\}$ is recognized as a q-length word $w = \{v_1, v_2, \ldots, v_q\}$ is defined as $P(w|s_1 s_2 \ldots s_q) = \prod_{i=1}^{q} P(v_i|s_i) CF(s_i)$.

$P(v_i|s_i)$ is the probability that s_i is recognized as v_i. It can be defined by the similarity that s_i is recognized as v_i. $CF(s_i)$ represents the confidence level of s_i, which can be defined by a measurement on the average CC spatial distance within s_i. Clearly, if the spatial distance among the CCs within s_i is big, s_i will be less likely to be character. We next define the probability that a word string $w_1, w_2, \ldots, w_p$ is recognized as an address $a = \{a_1, a_2, \ldots, a_n\}$. a_i represents the i-th address level such as $a_1 =$北京, $a_2 =$海淀区.

$$P(a_1, a_2, \ldots, a_p|w_1, w_2, \ldots, w_n) = \sqrt[L]{\prod_{i=1}^{p} P(a_i|w_i) CF(w_i)} \tag{3}$$

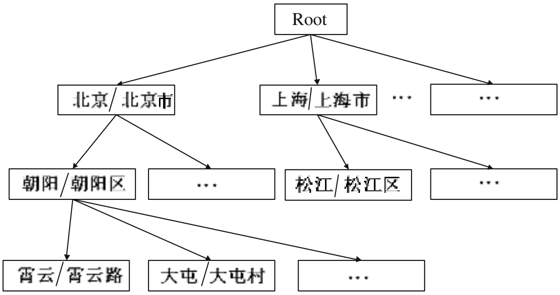


Fig. 4. Address Knowledge Tree

$$L = \sum_{i=1}^{p} length(a_i) \tag{4}$$

In the above, $P(a_i|w_i)$ is the probability that a word w_i is recognized as a word a_i; $length(a_i)$ represents the number of the characters contained in a_i; $CF(w_i)$ describes the probability that the associated segment string $\{s_1^i, s_2^i, \ldots, s_o^i\}$ is recognized as w_i, i.e., $CF(w_i) = P(w_i|s_1^i s_2^i \ldots s_o^i)$.

The WBLM aims at finding an address string $\{a_1, a_2, \ldots, a_p\}$ with the maximum value of $P(a_1 a_2 \ldots a_p|w_1 w_2 \ldots w_n)$. In practice, to speed up the calculation, the log form of this probability is usually adopted.

4 Combination

In this section, we discuss how to combine our Enhanced Holistic Word Matching approach with the Word Based Language Model. When there are no missing key characters, holistic word matching approach avoids segmenting each character one by one. This enables this approach an inherent advantage over other character-based methods. However, irregular Chinese address is also commonly seen in practice. In this case, the enhanced holistic word based approach will definitely fail. On the other hand, the word based language model can flexibly incorporate address knowledge; it does not depend on the extraction of key characters. Therefore, the word based language model is more suitable when key characters are missing. Taking account of the advantages of both models, we propose the following simple yet effective combination strategy. The address string is first input to the Enhanced Holistic Word Based Approach for recognition. If there are no missing key characters, the average word matching distance will be small. Otherwise, if some key characters are missing, the Enhanced Holistic Word Based Approach will force regarding some characters as key characters. Therefore the word matching will inevitably output large matching distance, since the word boundaries, i.e., the key characters, are not correctly obtained. Hence we can simply judge whether the average matching distance is greater than a threshold Th_1 or not so that we can determine whether the input address is regular or irregular. If it is regular, we output the answer given by the Enhanced Word Holistic Approach; otherwise, the final recognition is output by the Word Based Language Model.

5 Experiments

In this section, we evaluate our algorithm's performance against the plain recognition and the traditional holistic word matching approach. We first describe the data sets used in this paper briefly.

Three data sets are used to evaluate the performance of the new system. These data sets, which are of low, medium, and good quality respectively, consist of nearly 1800 images (around 300 regular address and 300 irregular images per data set). These images are written by different persons from different societies.

Note that we follow [7] and do not consider the address part after the last key character (this part might be the building name, room number etc). We use the string recognition rate (*SRR*) as the performance metric. The SRR is defined $SRR = \frac{\text{The number of correctly recognized address strings}}{\text{The total number of address strings}}$. An address string is regarded to be correctly recognized if and only if all the characters in this string accords with the ground truth.

In Table 1, we report the *SRR* performance of our proposed hybrid approach in comparison with the plain recognition approach (PRA) and the traditional holistic word approach (HWA). Our hybrid approach outperforms these two approaches distinctively. The PRA approach cannot appropriately deal with the difficulties of segmentation. Moreover, directly recognizing single handwritten Chinese characters presents a large-category pattern recognition task, which has been proved to be a very difficult problem. This leads to its low recognition rate. On the other hand, The HWA approach lacks the scheme to deal with irregular address, resulting in a definite failure in recognizing the address string with missing key characters. Since irregular address accounts for half of the test strings, the accuracy of HWA never surpasses 50%. In contrast, our proposed approach can deal with all the three difficulties that exist in HCAR and therefore naturally outperforms the other two in terms of the recognition accuracy.

Table 1. String Recognition Rate in three data sets

Dataset	Low Quality	Medium Quality	Good Quality
PRA(%)	0.91	4.74	29.44
HWA(%)	27.90	33.12	42.88
Our Appr.(%)	**80.97**	**86.32**	**90.52**

6 Conclusion

We have proposed a novel hybrid approach for Handwritten Chinese Address Recognition. This approach combines the enhanced holistic word matching approach with the word-based language model and has overcome all the three difficulties that cannot be solved by other traditional methods. Experiments on different quality and different types of address strings show that the proposed approach outperforms other traditional methods significantly.

References

1. R. Bloem, H. N. Gabow, and F. Somenzi. An algorithm for strongly connected component analysis in $n \log n$ symbolic steps. In W. A. Hunt, Jr. and S. D. Johnson, editors, *Formal Methods in Computer Aided Design*, pages 37–54. Springer-Verlag, November 2000. LNCS 1954.
2. Q. Fu, X. Ding, Y. Jiang, C. Liu, and X. Ren. A hidden markov model based segmentation and recognition algorithm for chinese handwritten address character strings. In *Eigth International Conference on Document Analysis and Recognition (ICDAR-2005)*, pages 590–594, 2005.

3. Joshua T. Goodman. A bit of progress in language modeling. *Computer Speech and Language*, 15:403–434, 2001.
4. Z. Han, C. P. Liu, and X. C. Yin. A two-stage handwritten character segmentation approach in mail address recognition. In *Eigth International Conference on Document Analysis and Recognition (ICDAR-2005)*, pages 111–115, 2005.
5. Y. Hotta, H. Takebe, and S. Naoi. Holistic word recognition based on synthesis of character features. In *Fourth IAPR International Workshop on Document Analysis Systems (DAS-2000)*, pages 313–324, 2000.
6. S. Naoi, M. Suwa, and Y. Hotta. Recognition of handwritten japanese addresses based on key character extraction and holistic word matching. In *Third IAPR International Workshop on Document Analysis Systems (DAS-1998)*, pages 149–152, 1998.
7. C. Wang, Y. Hotta, M. Suwa, and S. Naoi. Handwritten chinese address recognition. In *Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9)*, 2004.

A Morphological Neural Network Approach for Vehicle Detection from High Resolution Satellite Imagery

Hong Zheng[1], Li Pan[2], and Li Li[1]

[1] Research Center for Intelligent Image Processing and Analysis,
School of Electronic Information, Wuhan University
129 Luoyun Road, Wuhan, Hubei 430079, China
zhenghong@21cn.com
[2] School of Remote Sensing Information& Engineering,
Wuhan University, 129 Luoyun Road, Wuhan, Hubei 430079, China
li.pan@126.com

Abstract. This paper introduces a morphological neural network approach to extract vehicle targets from high resolution panchromatic satellite imagery. In the approach, the morphological shared-weight neural network (MSNN) is used to classify image pixels on roads into vehicle targets and non-vehicle targets, and a morphological preprocessing algorithm is developed to identify candidate vehicle pixels. Experiments on 0.6 meter resolution QuickBird panchromatic data are reported in this paper. The experimental results show that the MSNN has a good detection performance.

1 Introduction

With the development of traffic there is high demand in traffic monitoring of urban areas. Currently the traffic monitoring is implemented by a lot of ground sensors like induction loops, bridge sensors and stationary cameras. However, these sensors partially acquire the traffic flow on main roads. The traffic on smaller roads – which represent the main part of urban road networks – is rarely collected. Furthermore, information about on-road parked vehicle is not collected. Hence, area-wide images of the entire road network are required to complement these selectively acquired data. Since the launch of new optical satellite systems like IKONOS and QuickBird, this kind of imagery is available with 0.6-1.0 meter resolution. Vehicles can be observed clearly on these high resolution satellite images. Thus new applications like vehicle detection and traffic monitoring are raising up. This paper intends to study the vehicle extraction issue from high resolution satellite images.

Some vehicle detection methods have been studied using aerial imagery [1][2][3][4]. In the existing methods, two vehicle models are used. They are explicit model and appearance-based implicit model. The explicit model describes a vehicle as a box or wire-frame representation. Detection is carried out by matching the model "top-down" to the image or grouping extracted image features "bottom-up" to create structures similar to the model.

Few research on vehicle detection from high-resolution satellite imagery with a spatial resolution of 0.6-1.0m has been reported [5][6]. At 0.6-1.0 meter resolution,

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 99 – 106, 2006.

vehicle image detail is too poor to detect a vehicle by model approaches. Thus, it is necessary to develop specific approaches to detect vehicles from high resolution satellite imagery.

Morphological shared-weight neural network (MSNN) combines the feature extraction capability of mathematical morphology with the function-mapping capability of neural networks in a single trainable architecture. It has been proven successful in a variety of automatic target recognition (ATR) applications [7][8][9]. Automatic vehicle detection belongs to ATR research, thus, in this paper the MSNN is employed to detect vehicle targets.

In this paper, we concentrate the vehicle detection on roads and parking lots, which can be manually extracted in advance. In order to reduce searching cost and false alarm, a morphology based preprocessing algorithm is developed. The algorithm automatically identifies candidate vehicle pixels which include actual vehicle pixels and non-target pixels similar to vehicle pixels. Some of sub-images centered at those pixels are selected as the vehicle and non-vehicle training samples of the MSNN. The trained MSNN is tested on real road segments and parking lots. The performance results are also discussed in this paper.

The paper is organized as follows. In Section 2, the details of our vehicle detection approach are described. In Section 3, experimental results are given and conclusions are provided in Section 4.

2 Vehicle Detection Approach

The vehicle detection is carried out by an MSNN classification method. Before describing the vehicle detection approach, we briefly introduce the MSNN architecture as follows.

2.1 MSNN Architecture

Before describing the MSNN architecture, we provide brief definitions of some gray scale morphological operations. A full discussion can be found in [10]. The basic morphological operations of erosion and dilation of an image *f* by a structuring element (SE) *g* are

$$erosion:\ (f\Theta g)(x)=\min\left\{f(z)-g_x(z): z\in D\left[g_x\right]\right\} \tag{1}$$

$$dilation:\ (f\oplus g)(x)=\max\left\{f(z)-g^*_x(z): z\in D\left[g^*_x\right]\right\} \tag{2}$$

where $g_\psi(z) = g(z-x)$, $g(z) = -g(-z)$ and D[g] is the domain of g. The gray-scale hit-miss transform is defined as

$$f\otimes(h,m)=(f\Theta h)-(f\oplus m^*) \tag{3}$$

It measures how a shape *h* fits under *f* using erosion and how a shape *m* fits above *f* using dilation. High values indicate good fits.

MSNN is composed of two cascaded sub-networks: feature extraction (FE) sub-network and feed-forward (FF) classification sub-network. The feature extraction sub-network is composed of one or more feature extraction layers. Each layer is composed of one or more feature maps. Associated with each feature map, is a pair of structuring elements – one for erosion and one for dilation. The values of a feature map are the result of performing a hit-miss operation with the pair of structuring elements on a map in the previous layer (see Fig. 1). The values of the feature maps on the last layer are fed to the feed-forward classification network of the MSNN [11][12].

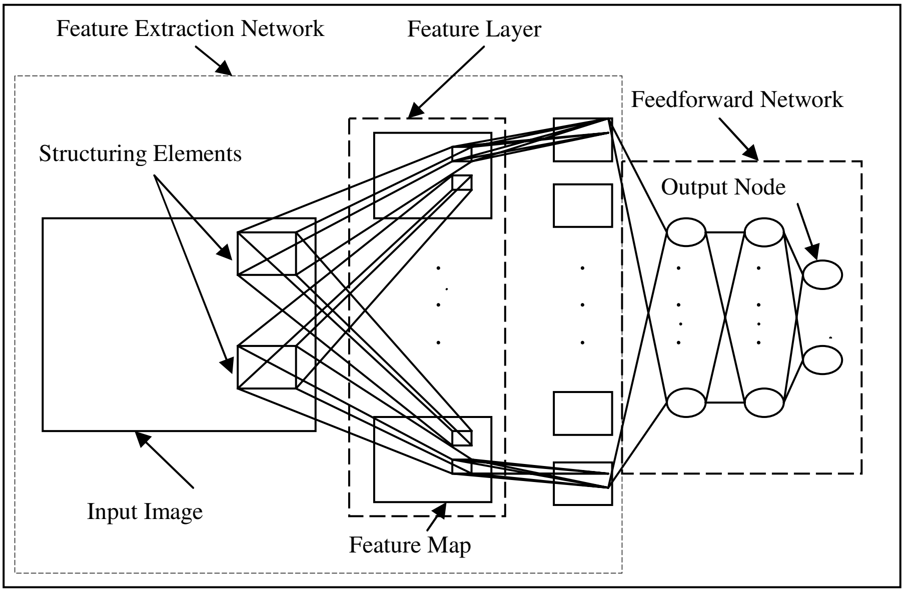


Fig. 1. The architecture of the morphology shared-weight neural network

2.2 Vehicle Detection Using MSNN

2.2.1 Morphology Preprocessing

In order to reduce searching cost and false alarm, a morphology based preprocessing algorithm is developed. In the algorithm, some morphological operations are used to enhance vehicle targets. These morphological operations are gray-scale top-hat and bottom-hat transforms, which are defined as

$$\text{top-hat: } T-HAT\left(f\right)=f-\left(f\circ g\right) \tag{4}$$

$$\text{bottom-hat: } B-HAT\left(f\right)=\left(f\bullet g\right)-f \tag{5}$$

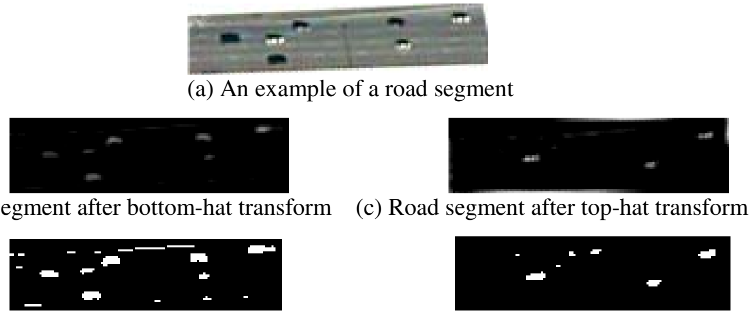
where $f\circ g$ and $f\bullet g$ means opening operation and closing operation respectively, i.e.

$$\text{opening: } f\circ g=\left(f\Theta g\right)\oplus g \tag{6}$$

$$\text{closing: } f\bullet g=-\left(\left(-f\right)\circ\left(-g\right)\right) \tag{7}$$

From empirical observation, the width of most vehicles on QuickBird images generally is less than or equal to 4 meters, and the length is not more than 6 meters. Thus the SE used is a disc with radius $r = 3$. Bright vehicles are smoothed out by the morphological opening operation and dark vehicles are smoothed out by the morphological closing operation. As a result, vehicles generally have a high value either on the top-hat image or the bottom-hat image. By setting a threshold on the top-hat image or the bottom-hat image, almost all vehicle pixels are detected and non-target pixels most similar to the vehicle pixels are also extracted. The threshold is obtained automatically using Ostu method [13]. In the Ostu method, pixels of a given image are represented in L gray Levels [1,2,...,L]. The number of pixels at level i is donated by n_i, and the total number of pixels by $N=n_1+n_2+...+n_L$. Then the dichotomisation of pixels into two classes C0 and C1, which denote respectively pixels with [1...k] and [k+1...L]. The method determines the threshold by determining the grey level that maximizes the between-class variance of the gray level histogram.

Fig.2(a) shows a road segment, and Fig.2(b)-(d) show its top-hat image, the bottom-hat image and their binary images after thresholding. From Fig.2(b)-(c), it can be seen that both bright vehicles and dark vehicles are enhanced after morphology preprocessing. As a result, these vehicles are labeled as white after thresholding. However, some noise like bright lane marks and tree shadow are also enhanced and mixed with vehicles. In order to further discriminate vehicle target pixels and non-vehicle target pixels, MSNN is introduced to implement pixel classification.



(a) An example of a road segment

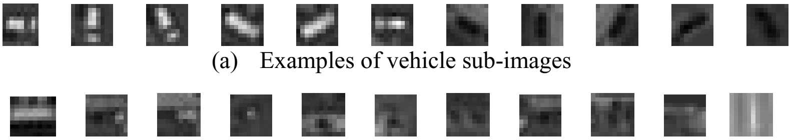
(b) Road segment after bottom-hat transform (c) Road segment after top-hat transform

(d) Thresholding result of road segment in (b) (e) Thresholding result of road segment in (c)

Fig. 2. An example of the morphology preprocessing algorithm

2.2.2 Network Training and Classification Testing

After the morphology preprocessing, the candidate vehicle pixels are obtained (see Fig.2(d)-(e)). Based on these candidate pixels, some sub-images centered at these pixels are selected as the vehicle and non-vehicle training samples of the MSNN. During training, test sub-images provide the input to the first feature extraction layer and the final output is a classification of "vehicle" or "non-vehicle". This method of training is called the "class-coded" mode of operation. While the network outputs values of 0 to 1 representing the confidence that an input represents a vehicle or non-vehicle, the returned result is an actual classification.



(a) Examples of vehicle sub-images

(b) Examples of non-vehicle sub-images

Fig. 3. Examples of training sub-images

Training data consists of a set of sub-images, which contain bright vehicles, dark vehicles, varying views of the "vehicle" and different "background". Fig.3 shows some examples of training sub-images.

Several parameters specify and/or affect network training. The regularization parameter indicates the reliability of the training set, with a value of zero indicating that the set is completely reliable and a value approaching infinity indicating less reliability. The learning rate and momentum constant are used to adjust the speed of convergence and stability while reaching a desired error size.

Weights for the feature extraction operation are user-initialized, while the initial feedforward weight matrices are populated by a random number generator. All FE and FF weights are learned by back propagation. A signal completes its forward pass and then the correction its backward pass at the end of each training epoch, before the next input begins processing. A weight correction is the function of the learning and momentum parameters, the local gradient of the activation function, and the input signal of the neuron.

After learning, the trained weights are used to implement pixel classification, which includes the feature extraction and feedforward classifications. Feature extraction is performed over the entire image rather than on a sub-image. The resulting feature maps centered at the candidate vehicle pixels with subimage-sized windows are input into the feedforward network for classification, and output value represents the attribution of the candidate vehicle pixel, i.e., vehicle pixel or non-vehicle pixel.

3 Experimental Results

QuickBird panchromatic data set used in our study was collected from Space Imaging Inc. web site. The data set contains different city scenes. A total of 15 road segments and 5 parking lots segments containing over 1000 vehicles were collected. Most vehicles in the images are around 5 to 10 pixels in length and around 3 to 5 pixels in width. Since the vehicles are represented by a few pixels, their detection is very sensitive to the surrounding context. Accordingly, the collected images consist of a variety of conditions, such as road intersections, curved and straight roads, roads with lane markings, road surface discontinuity, pavement material changes, shadows cast on the roads from trees, etc. These represent most of the typical and difficult situations for vehicle detection.

For each selected road segment image or parking lot image, roads and parking lots were extracted manually in advance and vehicle detection was performed only on the extracted road surfaces. To build the vehicle example database, a human expert manually delineated the rectangular outer boundaries of vehicles in the imagery.

A total of 100 vehicles delineated in this manner from 10 road segments. An image region with size 6×6m can cover most vehicles in the imagery. Hence, sub-images of size 10×10 pixels centered at vehicle centroids were built into the vehicle example database. Taking vehicle orientations into account, each sub-image was rotated every 45° and the resulting sub-images were also collected in the vehicle example database. As a result, the vehicle example database consisted of 100×4 = 400 sub-image samples. In addition, 400 non-vehicle sub-image samples covering different road surfaces were also collected to build the non-vehicle example database.

After building sample databases, sub-image samples were used to train the MSNN and validate the vehicle detection approach. The MSNN used in our experiments had a 20×20 input and one feature extraction layer with two feature maps. The downsampling rate was 2 (i.e., 10×10 feature maps) and the structuring elements were 5×5. The feed-forward network of the MSNN was composed of a two-node input layer, ten-node hidden layer and a two-node output layer (target and non-target). All weights were initialized with random numbers in [-0.1, 0.1]. The learning rate was 0.002. A logistic function was used as the activation function. The expected outputs for vehicle targets and non-vehicle targets were set to [1, 0] and [0, 1] respectively. With these training parameters, the network was trained for 1600 epochs.

After training, the MSNN was tested on 15 road segments and 5 parking lots. The detection statistical results are shown in Tables 1. Fig. 4 shows some images of vehicle detection results.

Table 1. Vehicle detection results

Site	No. of vehicles	No. of detected vehicles	No. of missing vehicles	No. of false alarm	Detection rate %
Road1	6	5	1	0	83.3
Road2	8	7	1	0	87.5
Road3	11	9	1	1	81.8
Road4	15	13	2	0	86.6
Road5	20	16	3	1	80
Road6	18	15	3	0	83.3
Road7	28	23	5	0	82.1
Road8	63	52	8	3	82.5
Road9	54	41	10	3	75.9
Road10	82	66	12	4	80.4
Road11	114	92	15	7	80.7
Road12	154	125	23	6	81.1
Road13	210	175	29	6	83.3
Road14	268	227	31	10	84.7
Road15	304	234	50	20	76.9
Parking1	7	5	2	0	71.4
Parking2	13	9	4	0	69.2
Parking3	20	13	5	2	65
Parking4	46	28	15	3	60.8
Parking5	90	46	40	4	51.1

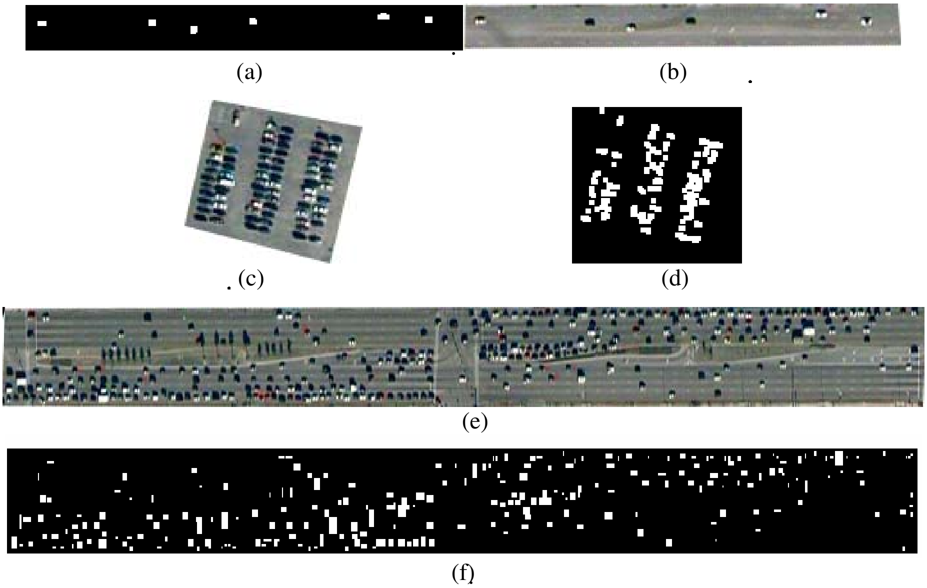


Fig. 4. Vehicle detection results (a)(c)(e) The original images of road segments and parking lots. (b)(d)(f) The binary images of vehicle detection results for images shown in (a)(c)(e).

From Table 1, it can bee seen that the detection rates (number of detected vehicles/number of vehicles) for road segments are from 75.9% to 87.5%, and average detection rate is 82%. The detection rates vary with the complexity of road surfaces, as well as the false alarm. The false alarms are due to vehicle-like "blobs" present in some of complex urban scenes, such as the presence of dust and lane markings (see Fig. 4). Some of these "blobs" are very hard to distinguish from actual vehicles, even to a trained eye. Most missing detections occur when the vehicles have a low contrast with the road surface or vehicles are too close.

For the vehicle detection on parking lots, the detection rates are not high. It is because the vehicles are too close to separate due to the resolution limit. How to detect vehicles on parking lots is still an open issue.

4 Conclusions

In this paper, we focus on the issue of vehicle detection from high resolution satellite imagery. We present a morphology neural network approach for vehicle detection from 0.6 meter resolution panchromatic QuickBird satellite imagery. A MSNN was introduced in our approach and was found to have good vehicle detection performance. Further work could include more training samples, better pre-processing method such as adaptive image enhancement and filtering, and introducing more information like edge shapes to improve the detection rate.

Acknowledgement

This work was supported by the National Nature Science Foundation of China (Grant No. 40571102).

References

[1] Ruskone R., Guigues L., Airault S., and Jamet O.: Vehicle Detection on Aerial Images: A Structural Approach. In: Proceedings of International Conference On Pattern Recognition, Vienna, Austria (1996) 900-904.

[2] Zhao T. and Nevatia R.: Car Detection in Low Resolution Aerial Image. In: Proceedings of International Conference on Computer Vision, Vancouver, Canada (2001) 710-717.

[3] Schlosser C., Reitberger J., and Hinz S.: Automatic Car Detection in High Resolution Urban Scenes Based on An Adaptive 3D-model. In: Proceedings of the 2nd GRSS/ISPRS Joint Workshop on Data Fusion and Remote Sensing over Urban Area, Berlin, Germany (2003) 167-170.

[4] Stilla U, Michaelsen E, Soergel U, Hinz S, Ender HJ.: Airborne Monitoring of Vehicle Activity in Urban Areas. In: Altan MO (ed) International Archives of Photogrammetry and Remote Sensing, 35(B3) (2004) 973-979.

[5] Sharma G.: Vehicle Detection and Classification in 1-m Resolution Imagery. Ohio State University, Master of Science thesis (2002).

[6] Gerhardinger A., Ehrlich D., Pesaresi M.: Vehicles Detection from Very High Resolution Satellite Imagery, In: Stilla U, Rottensteiner F, Hinz S (Eds), International Archives of Photogrammetry and Remote Sensing, Vol. XXXVI, Part 3/W24 (2005) 83-88.

[7] Won Y. : Nonlinear Correlation Filter and Morphology Neural Networks for Image Pattern and Automatic Target Recognition. Ph.D. Thesis, University of Missouri, Columbia, Miss (1995)

[8] Won Y, Gader PD, Coffield P.: Morphological Shared-Weight Networks with Applications to Automatic Target Recognition. IEEE Trans. on Neural Networks, 8 (1997) 1195–1203.

[9] Khabou M.A., Gader P.D. and Keller J.M.: LADAR Target Detection using Morphological Shared-weight Neural Networks. Machine Vision and Applications, 11 (2000) 300-305.

[10] Serra J.: Image Analysis and Mathematical Morphology, Vol. 2, Academic Press, New York, N.Y (1988).

[11] Gader PD, Won Y, Khabou MA. : Image Algebra Networks for Pattern Classification. In: Proceedings of SPIE Conference on Image Algebra and Morphological Image Processing, 2300 (1994) 157–168.

[12] Gader PD, Miramonti JR, Won Y, Coffield P.: Segmentation Free Shared-Weight Networks for Automatic Vehicle Detection. Neural Networks, 8 (1995) 1457–1473.

[13] Ostu N.: A Threshold Selection Method from Gray Level Histograms, IEEE Transactions on System, Management, Cybernet, 9 (1979) 62-66.

Secure Personnel Authentication Based on Multi-modal Biometrics Under Ubiquitous Environments

Dae-Jong Lee, Man-Jun Kwon, and Myung-Geun Chun*

Dept. of Electrical and Computer Engineering, Chungbuk National University,
Cheongju, Korea
mgchun@chungbuk.ac.kr

Abstract. In this paper, we propose a secure authentication method based on multimodal biometrics system under ubiquitous computing environments. For this, the face and signature images are acquired in PDA and then each image with user ID and name is transmitted via WLAN (Wireless LAN) to the server and finally the PDA receives authentication result from the server. In the proposed system, face recognition algorithm is designed by PCA and LDA. On the other hand, the signature verification is designed by a novel method based on grid partition, Kernel PCA and LDA. To calculate the similarity between test image and training image, we adopt the selective distance measure determined by various experiments. More specifically, Mahalanobis and Euclidian distance measures are used for face and signature, respectively. As the fusion step, decision rule by weighted sum fusion scheme effectively combines the two matching scores calculated in each biometric system. From the real-time experiments, we convinced that the proposed system makes it possible to improve the security as well as user's convenience under ubiquitous computing environments.

1 Introduction

With the advance in communication network, the electronic commerce has been popular according to the rapid spread of Internet. In particular, wireless devices make it possible to enrich our daily lives in ubiquitous environments. Network security, however, is likely to be attacked with intruders and it is faced with the serious problems related to the information security. It is even more difficult to protect the information security under the wireless environment comparing with the wired network. One of the most conventional methods for system security is using password, which is very simple and does not require any special device. However, it can be easily divulged to others. To tackle these problems, biometrics is emerging as a promising technique. In the biometrics, a number of researchers have studied iris, facial image, fingerprint, signature, and voiceprint. Among them, the face recognition is known as the most natural and straightforward method to identity each person.

This face recognition has been studied in various areas such as computer vision, image processing, and pattern recognition. Popular approaches for face recognition

* Corresponing author.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 107 – 115, 2006.

are PCA (Principle Component Analysis) [1] and LDA (Linear Discriminant Analysis) [2] methods. However, the major problem with the use of above methods is that they can be easily affected by variations of illumination condition and facial expression. One the other hand, the signature has been a familiar means where it is used for a personal authentication such as making a contact. Online signature recognition methods roughly belong to one of global feature comparison, point-to-point comparison and segment-to-segment comparison methods [3]. For a signature, however, comparing with other features of biometrics, its skilled forgery is more or less easy and system performance is often deteriorated by signature variation from various factors [4].

Though advanced researches based on single biometric modality have been proposed for information security, there are some problems to apply in real life because of lack of confidential accuracy [5-7]. Multimodal biometrics has the advantage of improving security by combination of two biometric modalities such as face and signature [8]. In this paper, we describe an implementation of multimodal biometrics under ubiquitous computing environments. The proposed system is implemented with embedded program in PDA. More specifically, the face images and signatures images are obtained by PDA and then these images with user ID and name are transmitted via WLAN (Wireless LAN) to the server and finally the PDA receives verification result from the server. In our system, face verification system is implemented by conventional PCA and LDA method which calculates eigenvector and eigenvalue matrices using the face image from the PDA at enrollment steps. The signature verification is designed by a novel method based on grid partition, Kernel PCA and LDA. To calculate the similarity between test image and training image, we adopt the selective distance measure determined by various experiments. Here, Mahalanobis and Euclidian distance measures are used for face and signature, respectively. As the fusion step, decision rule by weighted sum fusion scheme is used to effectively combine the two matching scores calculated in each biometric system. The implemented system renders improvements of speed and recognition rate to increase security under ubiquitous computing environments.

This paper is organized as follows. Section 2 describes the system architecture implemented in PDA. In Section 3, we describe the authentication methods for face and signature and fusion method. In Section 4, we presents experiment results obtained by real-time experiment. Finally, some concluding remarks are given in Section 5.

2 System Architecture for PDA Based Personnel Authentication

The proposed system consists of a client module for biometric data acquisition and a server module for authentication. The client module is to register the face and signature image. After then, the acquired user's information is transmitted via WLAN to the server which performs analyzing the transmitted data and sending the authentication result to the client module such as acceptance or rejection. That is, the server deals with image processing and verification algorithm.

For the client module, face images and signatures images are acquired in the PDA program implemented by Microsoft embedded development tool. In the face acquisition process, user's image is obtained from the camera attached to the PDA. Face detection is essential process since the recognition performance directly depends on the quality of acquired face image. Here, we capture a face image by considering the direction and position between two eyes. And then, the captured image is saved in 240x320 pixels BMP format. On the other hand, user's signature is acquired from PDA by stylus pen in acquisition process. Fig 2 shows the user interface environment to acquire the face and signature images. User may select the ID, name, and process step for identity or register from the setup menu.

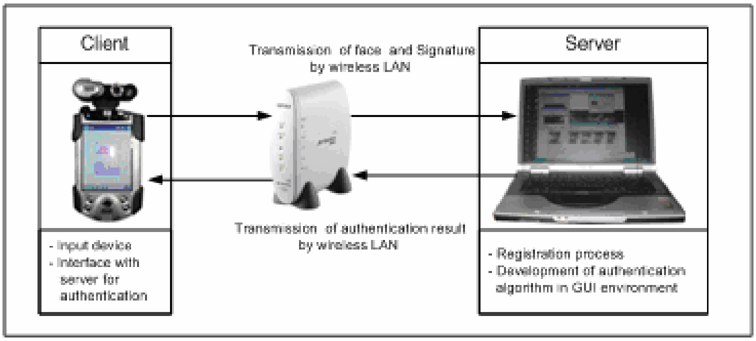
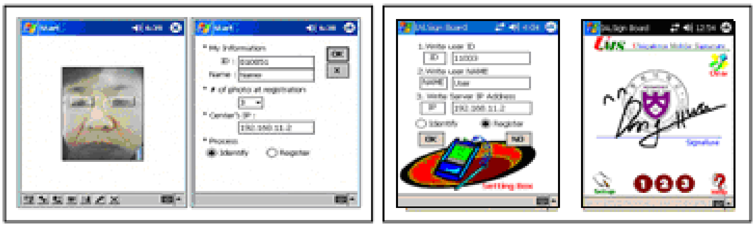

Client
Transmission of face and Signature by wireless LAN
Server
Transmission of authentication result by wireless LAN
- Input device
- Interface with server for authentication
- Registration process
- Development of authentication algorithm in GUI environment

Fig. 1. System architecture for biometric authentication



(a) Face

(b) signature

Fig. 2. User interface environment

In the server module, matching score is calculated between input image and registered images in the database. In the registration step for face recognition, feature extraction is performed by PCA and LDA for images transmitted via WLAN from the camera attached to the PDA. And then the calculated features are registered in the database. Authentication is performed by comparing input face with registered face images. For the signature authentication, server receives user's signature from PDA. And then feature extraction is performed by using the algorithm based on grid partition, Kernel PCA and LDA. Finally, decision making for acceptance is performed according to matching scores obtained face and signature module in server, respectively.

3 Multi-modal Biometric Authentication Algorithm

The proposed multi-modal biometric system consists of a face recognition module, a signature recognition module, and a decision module as shown in Fig. 3. Here, the face recognition is designed by PCA and LDA method. The signature recognition is implemented by the grid partition, Kernel PCA, and LDA. As a final step, decision module is implemented with the weighted sum rule. The face recognition system is composed of feature extraction and classification process parts. First, face image is decomposed in each frequency band by wavelet transform to compress it [5]. Then, PCA is applied to reduce the dimensionality of image for low frequency band. Here, we briefly describe the feature extraction based on PCA and LDA used in the face recognition.

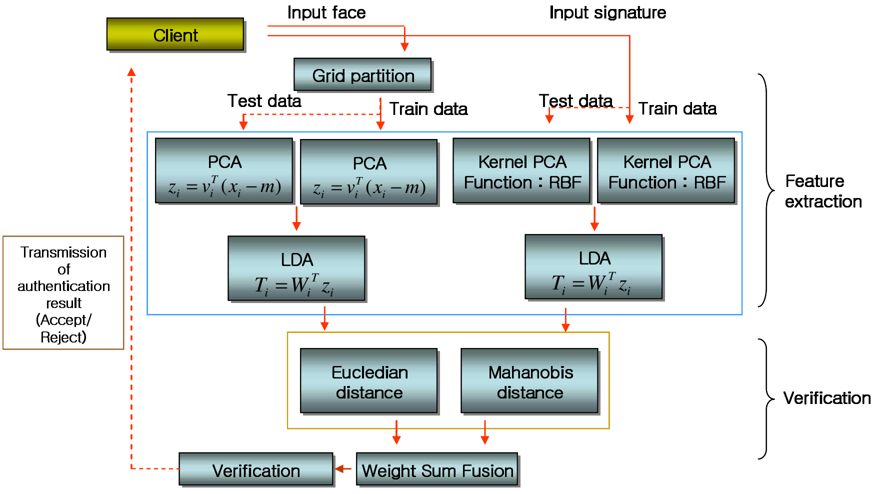


Fig. 3. Proposed multi-modal biometric system

Let a face image be a two-dimensional $n \times n$ array of containing levels of intensity of the individual pixels. An image $\mathbf{z}_i$ may be conveniently considered as a vector of dimension n^2. Denote the training set of N face images by $Z = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N)$. We define the covariance matrix as follows

$$R = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{z}_i - \bar{\mathbf{z}})(\mathbf{z}_i - \bar{\mathbf{z}})^T = \Phi\Phi^T \quad (1)$$

$$\bar{\mathbf{z}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{z}_i \quad (2)$$

Then, the eigenvalues and eigenvectors of the covariance matrix R are calculated, respectively. Let $E = (\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_r)$ denote the r eigenvectors corresponding to the r largest eigenvalues. For a set of original face images Z, their corresponding reduced feature vectors $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ can be obtained as follows;

$$\mathbf{x}_i = E^T(\mathbf{z}_i - \bar{\mathbf{z}}) \tag{3}$$

The second processing stage is based on the use of the LDA as follows. Consider c classes in the problem with N samples; let the between-class scatter matrix be defined as

$$S_B = \sum_{i=1}^{c} N_i(\mathbf{m}_i - \bar{\mathbf{m}})(\mathbf{m}_i - \bar{\mathbf{m}})^T \tag{4}$$

where N_i is the number of samples in i'th class C_i and $\bar{\mathbf{m}}$ is the mean of all samples, $\mathbf{m}_i$ is the mean of class C_i. The within-class scatter matrix is defined as follows

$$S_W = \sum_{i=1}^{c} \sum_{x_k \in C_i} (\mathbf{x}_k - \mathbf{m}_i)(\mathbf{x}_k - \mathbf{m}_i)^T = \sum_{i=1}^{c} S_{W_i} \tag{5}$$

where, S_{W_i} is the covariance matrix of class C_i. The optimal projection matrix W_{FLD} is chosen as the matrix with orthonormal columns that maximizes the ratio of the determinant of the between-class matrix of the projected samples to the determinant of the within-class fuzzy scatter matrix of the projected sampled, i.e.,

$$W_{FLD} = \arg\max_{W} \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|} = \left[\mathbf{w}_1 \quad \mathbf{w}_2 \quad \cdots \quad \mathbf{w}_m\right] \tag{6}$$

where $\{\mathbf{w}_i \mid i = 1,2,\cdots,m\}$ is a set of generalized eigenvectors (discriminant vectors) of S_B and S_W corresponding to the $c-1$ largest generalized eigenvalues $\{\lambda_i \mid i = 1,2,\cdots,m\}$, i.e.,

$$S_B \mathbf{w}_i = \lambda_i S_W \mathbf{w}_i \quad i = 1,2,\ldots,m \tag{7}$$

Thus, the feature vectors $V = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N)$ for any face images $\mathbf{z}_i$ can be calculated as follows

$$\mathbf{v}_i = W_{FLD}^T \mathbf{x}_i = W_{FLD}^T E^T(\mathbf{z}_i - \bar{\mathbf{z}}) \tag{8}$$

After obtaining the feature vectors, the classification is achieved by finding the minimum distance between the coefficients of test patterns and training patterns. Here, the distance is calculated by Mahalanobis distance measure.

For the signature recognition system, features are calculated by Kernal PCA and LDA. Before projecting the original features by Kernel PCA, a signature image is projected to vertical and horizontal axes by grid partition method [8]. Kernel PCA can be derived using the known fact that PCA can be carried out on the dot product matrix instead of the covariance matrix [9]. Let $\{x_i \in R^M\}_{i=1}^N$ denote a set of data. Kernel PCAfirst maps the data into a feature space F by a function $\Phi : R^M \to F$, and then performs standard PCA on the mapped data. Defining the data matrix X by $X = [\Phi(x_1)\ \Phi(x_2) \cdots \Phi(x_N)]$, the covariance matrix C in F becomes

$$C = \frac{1}{N}\sum_{1}^{N} \Phi(x_i)^T \Phi(x_i) = \frac{1}{N} X^T X \tag{9}$$

We assume that the mapped data are centered as $1/N \cdot \Sigma_1^N \Phi(x_i) = 0$. We can find the eigenvalues and eigenvectors of C via solving the eigenvalues problem

$$\lambda u = Ku \tag{10}$$

The $N \times N$ matrix K is the dot product matrix defined by $K = 1/N \cdot X^T X$ where

$$K_{ij} = \frac{1}{N}\Phi(x_i) \bullet \Phi(x_i) = \frac{1}{N}k(x_i, x_j) \tag{11}$$

Let $\lambda \geq \cdots \geq \lambda_p$ be the nonzero eigenvalues of K ($P \leq N$, $P \leq M$) and $u^1, \cdots, u^P$ the corresponding eigen-vectors. Then C has the same eigenvalues and there is a one-to-one correspondence between the nonzero eigen-vectors $\{u^h\}$ of K and the nonzero eigenvectors $\{v^h\}$ of C: $v^h = \alpha^h X u^h$, where α^h is a constant for normalization. If both of the eigenvectors have unit length, $\alpha^h = 1/\sqrt{\lambda_h N}$. We assume $\|v^h\| = 1/\sqrt{\lambda_h N}$ so that $\alpha^h = 1$.

For a test data x, its h^th principal component y_h can be computed using Kernel function as

$$y_h = v^h \bullet \Phi(x) = \sum_{i=1}^{N} u_i^k k(x_i, x) \tag{12}$$

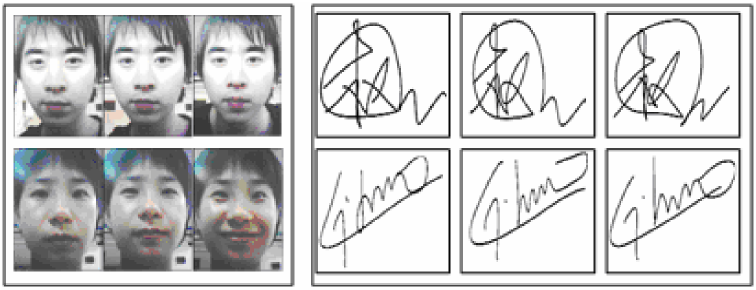
Then the Φ image of x can be reconstructed from its projections onto the first H ($\leq P$) principal components in F by using a projection operator P_H

$$P_H \Phi(x) = \sum_{h=1}^{H} y_h v^h \tag{13}$$

The Kernel PCA allows us to obtain the features with high order correlation between the input data samples. In nature, the Kernel projection of data sample onto the Kernel principal component might undermine the nonlinear spatial structure of input data. Namely, the inherent nonlinear structure inside input data is reflected with most merit in the principal component subspace. To extract feature, we use LDA as well as a Kernel PCA so as to examine the discriminative ability of Kernel principal components. After obtaining the feature vectors, classification is achieved by finding the minimum distance between the coefficients of test patterns and training patterns. Here, the distance is calculated by Euclidean distance measure and finally fusion scheme is implemented by weighted sum rule for similarities obtained from face and signature [8].

4 Experiments and Analysis

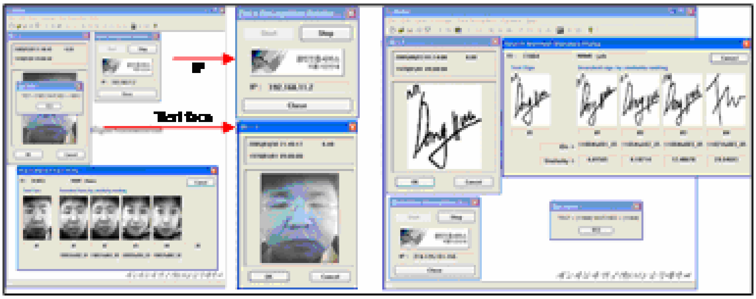
To evaluate the proposed method, face and signature are transmitted via WLAN from PDA. First, three faces and signatures for each user are registered with ID number. Recognition is performed by comparing face and signature images with registered ones. Fig. 4 shows some samples of face or signature images acquired from the PDA. The original size of face image is 240×320. However, it is resized as 128×128 pixel image whose gray level ranges between 0 and 255. Finally, the compressed face image is obtained by performing 4-level wavelet packet transform. On the other hand, the size of signature is 240×100. After applying the PPP matching, 2-dimensional signature is rearranged in vector form having horizontal and vertical information [8]. For the preprocessed face and signature images, feature extraction method is applied to obtain the features such as described in Section 3.



(a) Faces (b) Signatures

Fig. 4. Some samples of faces and signatures acquired from PDA

Figure 5 shows the recognition result executed in the server. As seen in Fig 5, four candidate images are displayed according to the matching score. In this Figure, the leftmost image is the testing one to be authenticated and the others are matched in the server. Among theses images, leftmost image has the highest matching score for the input image. So the server considers him as a genuine person when the matching score is higher than a predefined threshold value.

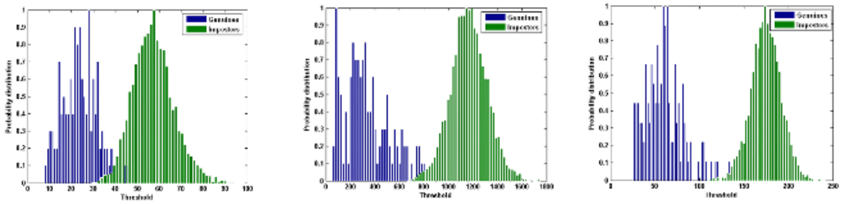


(a) Face recognition (b) Signature recognition

Fig. 5. Recognition process in the server

Figure 6 shows the similarity between genuine and imposter according to the threshold which determines the accept/reject for face and signature, respectively. Here, the number of data is 3 and 44 per person for genuine and imposter. Our experiment uses the 45 person. As seen in Fig. 6-(a) (b), performance shows the best performance when threshold is 38 and 780 for signature and face, respectively. Fig 6-(c) shows the performance applying fusion technique based on weighted sum rule. Final matching degree is calculated by d1+0.1×d2, where d1 is the value obtained

Euclidian distance for signature and d2 is the Mahalanobis distance for face. As seen in Fig. 6-(c), the fusion scheme makes discrimination between genuine and imposter larger than single biometrics.



(a) signature (b) face (c) fusion scheme

Fig. 6. Discrimination between genuine and imposter

Fig 7 shows the ROC curve representing FAR(false acceptance rate) and FRR(false reject rate). As seen in Fig 7, the performance shows best performance when threshold is 122. Table 1 shows recognition rate with respect to EER (error equal rate). The error rates are 5.5% and 3.7% by signature and face verification system, respectively. Finally, the multi-modality makes the error rate lower than single modal system by effectively combining two biometric features. More specifically, the error rate shows 1.1 % and one can find that the proposed method can be used to establish a higher security system.

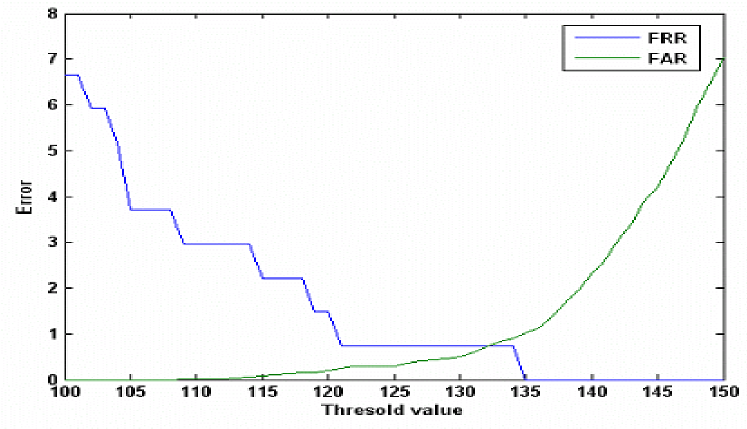


Fig. 7. The ROC curve obtained by applying fusion scheme

Table 1. EER rate according to each applied method

Rate	Signature	Face	Multi-modal
EER	5.5%	3.7%	1.1%

5 Concluding Remarks

In this work, we suggested a multimodal biometrics system under ubiquitous computing environments. Our system consists of face and signature verification system implemented in PDA and server network. Specifically, the face and signature are transmitted to server and PDA receives the verification results via WLAN coming from decision in the server. Face verification system is implemented by conventional PCA and LDA method and signature verification is designed by a novel method based on grid partition, Kernel PCA and LDA. As the fusion step, decision rule by weighted sum fusion scheme is used to effectively combine the two matching scores calculated in each biometric system. From various real-time experiments, we found that the fusion scheme made the error rate lower than single modal system. Therefore, we confirm that the proposed method can be applied to the applications for personnel authentication where higher security is required.

Acknowledgement. This work was supported by ETRI, Korea 2006.

References

[1] M. Turk, A. Pentland, Eigenfaces for Recognition, Journal of Cognitive Neuroscience, Vol. 3 (1991) 72-86

[2] Wenyi Zhao, Arvindh Krishnaswamy, Rama Chellappa, Discriminant Analysis of Principal Components for Face Recognition,Face Recognition from Theory to Application, Springer, (1998).

[3] Kiran G. V., Kunte R. S. R., Saumel S., On-line signature verification system using probabilistic feature modeling, Signal Processing and its Applications, Sixth International Symposium, Vol. 1 (2001) 351-358.

[4] Ma Mingming, Acoustic on-line signature verification based on multiple models, Computational Intelligence for Financial Engineering, Proceedings of the IEEE/IAFE/INFORMS Conference (2000) 30-33

[5] Keun-Chang Kwak, Pedrycz, W., Face Recognition using Fuzzy Integral and Wavelet Decomposition Method, Systems, Man and Cybernetics, Part B, IEEE Trans., Vol. 34 (2004) 1666-1675

[6] Jie Yang, Xilin Chen, Willam Junz, A PDA-based Face Recognition System, Proceeding of the sixth IEEE Workshop on Application of Computer Vision (2002) 19-23

[7] Jong Bae Kim, A Personal Identity Annotation Overlay System using a Wearable Computer for Augmented Reality, Consumer Electronics, IEEE Trans., Vol. 49 (2003) 1457 – 1467

[8] Dae Jong Lee, Keun Chang Kwak, Jun Oh Min, Myung Geun Chun, Multi-modal Biometrics System Using Face and signature, A.Lagana et al.(Eds.): LNCS 3043 (2004) 635-644

[9] B. Scholkopf, A. Smola, Nonlinear Component Analysis as a Kernel Eigenvalue Problem, Neural Computation, Vol. 10 (1998) 1299-1319

Pattern Classification Using a Set of Compact Hyperspheres

Amir Atiya[1], Sherif Hashem[2], and Hatem Fayed[2]

[1] Computer Engineering Department, Cairo University, Egypt
amiratiya@link.net
[2] Engineering Mathematics and Physics Department, Cairo University, Egypt
shashem@ieee.org, h_fayed@hotmail.com

Abstract. Prototype classifiers are one of the simplest and most intuitive approaches in pattern classification. However, they need careful positioning of prototypes to capture the distribution of each class region. Classical methods, such as learning vector quantization (LVQ), are sensitive to the initial choice of the number and the locations of the prototypes. To alleviate this problem, a new method is proposed that represents each class region by a set of compact hyperspheres. The number of hyperspheres and their locations are determined by setting up the problem as a set of quadratic optimization problems. Experimental results show that the proposed approach significantly beats LVQ and Restricted Coulomb Energy (RCE) in most performance aspects.

1 Introduction

The simplest and most intuitive approach in pattern classification is based on the concept of similarity [1]. Patterns that are similar (in some sense) are assigned to the same class. Prototype classifiers are one major group of classifiers that are based on similarity. A number of prototypes are designed so as they act as representatives of the typical patterns of a specific class. When presenting a new pattern, the nearest prototype determines the classification of the pattern. Two extreme ends of the scale for prototype classifiers are the nearest neighbor (NN) classifier, where each pattern serves as a prototype, and the minimum distance classifier, where there is only one prototype (the class center or mean) per class. Practically speaking, the most successful prototype classifiers are the ones that have a few prototypes per class, thus economically summarizing all data points into a number of key centers. Hyperspherical prototypes were first proposed in [2], in which, hyperspheres are used to represent each class region. This approach borrows ideas from what is called Restricted Coulomb Energy network classifiers. At each moment the system keeps some of the data points which were presented to it before, called prototypes together with a hypersphere centered around it. If a new point presented to the system is contained in some hyperspheres of inappropriate classes, the radii of the containing hyperspheres are reduced, so that none of them contains the new point. If the new point is contained in any of the hyperspheres of its own class, then no action is taken. If it is not contained in any hypersphere, then it becomes a new prototype, and is given its own hypersphere of some initial radius. Another improved version of RCE

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 116 – 123, 2006.

network was proposed by [3], namely RCE-2. In this method, only the hypersphere of the closest stored pattern from a different class is modified. In [4], two modifications are proposed to the standard RCE: (1) Assigning two thresholds for the hidden units that produce two hyperspheres and determine regions of rejections, (2) Modifying the center of the hidden unit towards newly presented training examples. Another variant of RCE is proposed in [5], where each class region is covered by a set of ellipsoids whose orientation coincides with the local orientation of the class region. The general learning scheme is similar to RCE's scheme that was suggested in [2]. Our proposed method, even though based on the concept of hyperspheres, is very different from the RCE approaches and their variants. For example, the RCE method are sequential in nature, while in our methods we consider all data points as a whole, allowing us to pose the problem as an optimization problem. The sequential nature of RCE, while giving it an adaptive nature, presents a problem for batch design in that the order of pattern presentation can have a big influence on the resulting final solution. As we will see in the simulation results, our method results in significantly less number of hyperspheres (hence more compact representation) and considerably better performance than that of RCE method. A preliminary version of the algorithm was introduced in [6].

2 Smallest Covering Hyperspheres

Consider a K-class pattern classification problem with N data points in a d-dimensional feature space. In the first proposed method the data points of each class are considered separately, and a number of hyperspheres as compact as possible are designed to cover the data points of the considered class (say class k). To achieve that, we first obtain the smallest hypersphere that encompasses all points of class k. This is derived by posing the problem as a quadratic optimization problem, as follows:

Given N data points in a d-dimensional feature space. Denote the data points as $\mathbf{a}_i = [a_{i1}, a_{i2}, \ldots, a_{id}]^T$, where i indexes the data point number. The problem is to find the center $\mathbf{x} = [x_1, x_2, \ldots, x_d]^T$ and radius r such that:

$$\text{Min } R \qquad \text{s.t.} \quad \|\mathbf{x} - \mathbf{a}_i\|_2^2 \le R \qquad , i=1,2,\ldots N \tag{1}$$

where $R=r^2$. The standard Lagrangian dual is:

$$\text{Min } \boldsymbol{\lambda}^T \mathbf{A}^T \mathbf{A} \boldsymbol{\lambda} - \mathbf{c}^T \boldsymbol{\lambda} \tag{2}$$

$$\text{s.t.} \quad \mathbf{e}^T \boldsymbol{\lambda} = 1 \qquad , \qquad \boldsymbol{\lambda} \ge 0$$

where $\mathbf{A} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \cdots \quad \mathbf{a}_N]$, $\mathbf{c} = [\mathbf{a}_1^T\mathbf{a}_1 \quad \mathbf{a}_2^T\mathbf{a}_2 \quad \cdots \quad \mathbf{a}_N^T\mathbf{a}_N]^T$, $\mathbf{e} \in R^N$ is the vector of all ones. This is a convex optimization problem and can be easily solved by any interior-point algorithm [7] or by the iterative barycentric coordinate descent method used in [8].

Next, we shrink the radius of the hypersphere to expel out all data points from different classes that happen to fall in the hypersphere. We now have a hypersphere that contains a number of points only from class k. We have now accounted for these points, so we remove them and tackle the remaining points of class k. We repeat the same procedure for these points and keep adding hyperspheres until we have covered all points from class k. We repeat the whole procedure for all other classes. The algorithm may be briefly described as follows:

Algorithm (SCHS)
Input:

- Training patterns pairs $\{\mathbf{a}_j, C(\mathbf{a}_j)\}$, j=1,2,…N where $C(\mathbf{a}_j) \in \{1,2,\ldots K\}$is the class index for pattern $\mathbf{a}_j$ and K is the number of classes.
- Minimum number of patterns to be encompasses by a hypersphere (N_{min}).
- Threshold $\mu \in [0,1]$.
- Threshold $\gamma \in [0,1]$.

Output: H hyperspheres with their corresponding classes.
Method:

1. Set H = 0, k = 1.
2. Set PointSet = all set of points of class k, Set H=H+1, TempSet=PointSet.
3. Find the smallest hypersphere (H) that encompasses the points in TempSet.
4. If number of samples of other classes encompassed by the hypersphere/ number of samples of class k encompassed by the hypersphere < μ then, go to step 9.
5. Find the farthest point from the center (**y**), whose class is different from k and is encompassed by the hypersphere. Compute its distance (d_y) from the center.
6. Drop points from TempSet, whose distances from the center are greater than or equal to d_y.
7. Repeat steps 3-6 until there are no points that need to be dropped in step 6.
8. If number of samples encompassed by the hypersphere < N_{min} then remove these samples, remove hypersphere H, set H=H-1, go to step 11.
9. If number of samples encompassed by the hypersphere/number of samples of the TempSet, then split the TempSet points into two groups (this will be illustrated later), remove hypersphere H, set H=H-1, tackle the points of both groups using the same procedure recursively by setting TempSet to be the points of the group under consideration and running from step 3.
10. Remove points encompassed by hyperspheres assigned from TempSet.
11. If PointSet is not empty, set TempSet=PointSet and go to step 3.
12. Else: If k < K, Set k = k + 1, go to step 2.

Another modification may be made to Step 6, is to drop the points from TempSet, whose distances from the center are greater than or equal to ηd_y rather than d_y, where $0<\eta\leq 1$. This modification gives the points close to the surface of the sphere the chance to be assigned to a perhaps better hypersphere and also speeds up the determination of the hypersphere (steps 3-6).

During the estimation of a hypersphere, sometimes the resultant hypersphere may encompass too few points of the PointSet. This case often occurs when samples of the class under consideration are clustered into groups separated by samples of other classes, or when the hypersphere center falls in a region of class overlap. In the next time step (of obtaining the next hypersphere) we will not be much better off, because not much have changed since only a few points have been chipped off from TempSet.

The outcome of this is that we end up with more hyperspheres than necessary. To overcome this standoff, the samples of the TempSet are divided into two groups (step 9) as follows:

Consider the direction of maximum variation of the points of PointSet, that is the first principal component, say α. Let $\bar{\mathbf{a}}$ be the mean vector of the points in PointSet. Then break these points into the two groups: $\mathbf{z}_\mathbf{i}^\mathbf{T}\alpha \geq 0$ and $\mathbf{z}_\mathbf{i}^\mathbf{T}\alpha < 0$ where $\mathbf{z}_\mathrm{i} = \mathbf{a}_\mathrm{i} - \bar{\mathbf{a}}$.

3 Classification Stage

Assume that the design of the hyperspheres is complete, and that it is required to classify a given data point (q). Several possibilities exist, for example the data point could fall inside several hyperspheres, or it could fall outside all hyperspheres. We have chosen to use the distance to the outside surface of the hypersphere (with that distance counted as negative if the point is inside the hypersphere) as the selection criterion. Specifically, we perform the following steps:

1. Compute the distance (d_i) between the data point and the center of each hypersphere H_i.
2. The index for nearest neighbor hypersphere I_q is chosen as:

$$I_q = \underset{i =\in\{1,2,3,...H\}}{\arg\min} (d_i - r_i) \tag{3}$$

where H is the total number of hyperspheres, r_i is the radius of hypersphere H_i.

4 Experimental Results

To validate our methods, we used two synthetic data sets and three real world data sets. In our implementation we used MATLAB 6.5 on Windows XP operating system running on Intel PC 2.4GHZ 256MB RAM. We compared our results with the learning vector quantization LVQ [9], RCE [2] and RCE-2 [3]. Best parameters for each method are selected using 5-fold cross validation [10]. In LVQ, the initial value of the learning rate (ε_0) is one of the following {0.01,0.05,0.1} and is decreased as a proportional to the reciprocal of the iteration number; i.e. $\varepsilon_i = \varepsilon_0 / i$ where i is the iteration number. The number of prototypes (P_k) for class k is selected as: $\delta \times N_k$ where $\delta \in \{0.01,0.1,0.2\}$, N_k is the number of training patterns whose class is k. Since all the training data used in our experiments are already randomized, we select the first P_k patterns for each class as the initial positions of the prototypes. For RCE networks, suggested values for the minimum radius allowed, r_{min} are: $\varepsilon \times \min_j(\max_i(a_{ij}) - \min_i(a_{ij}))$ where i=1,2,…N and j=1,2,…d and $\varepsilon \in \{0.001,0.01,0.1\}$ while those for the initial radius, r_{max} are: $\delta \times \min_j(\max_i(a_{ij}) - \min_i(a_{ij}))$ where i=1,2,…N and j=1,2,…d and $\delta \in \{0.5,0.75,1\}$. For SCHS, suggested values for μ are: 0.01,0.05,0.1, suggested values for γ are: 0.5,0.75, suggested values for N_{min} are: $\delta \times N_k$

where $\delta \in \{0.01, 0.05, 0.1\}$ and suggested values for η are:0.9,0.95. We now describe the data sets used.

4.1 I-I Data Set (I-I)

This is a two-class, 6-dimensional problem generated from normal distributions $N(\mu_i, \Sigma_i)$, i =1,2 [11]. The parameters are: $\mu_1 = [0\ 0\ \ldots\ 0]^T$, $\mu_2 = [\ 2.56\ 0\ \ldots\ 0]^T$, $\Sigma_1 = \Sigma_2 = I$. The value of μ controls the degree of overlap between the two distributions. 1,000 examples were used for training and 10,000 for testing.

4.2 Parabolic Boundary Data Set (Parabola)

This data set is generated as follows: 11,000 uniformly randomly distributed points were generated in two dimensions and two classes are assigned according to the following predetermined parabolic boundary:

$$\begin{array}{ll} (x-y)^2 - \sqrt{2}(x+y) + 1 > N(0, 0.05^2) & \text{class 1} \\ \text{Otherwise} & \text{class 2} \end{array} \tag{4}$$

where both x and y $\in$ [0,1] and $N(0, 0.05^2)$ denotes a number generated from a normal distribution with zero mean and standard deviation of 0.05. This allows some class overlap at the boundary, which is typically expected in the majority of pattern classification problems. 1,000 points were used for training and 10,000 for testing.

4.3 Thyroid Data Set

This data set is about the diagnosis of thyroid hypofunction. Based on patient query data and patient examination data, the task is to decide whether the patient's thyroid has overfunction, normal function, or underfunction. The data set consists of 21 inputs, 1 discrete output, 7200 examples. The class probabilities are 5.1%, 92.6% and 2.3% respectively. 5400 examples were used for training and the remaining 1800 examples were used for testing. This data set was obtained from thyroid1.dt file from Proben1 database [12] (which was created based on the "ann" version of the "thyroid disease" problem data set from the UCI repository of machine learning databases).

4.4 Satimage Data Set

The original Landsat data for this database was generated from data purchased from NASA by the Australian Centre for Remote Sensing, and used for research at the University of New South Wales. The sample database was generated taking a small section (82 rows and 100 columns) from the original data. The database is a (tiny) sub-area of a scene, consisting of 82×100 pixels, each pixel covering an area on the ground of approximately 80×80 meters^2. The information given for each pixel consists of the class value and the intensities in four spectral bands. Two of these are in the visible region (corresponding approximately to green and red regions of the visible spectrum) and two are in the (near) infra-red. Information from the neighborhood of a pixel might contribute to the classification of that pixel, the spectra of the eight neighbors of a pixel were included as attributes together with the four spectra of that pixel. Each line of data corresponds to a 3×3 square neighborhood of pixels completely contained within the 82×100 sub-area. Thus each line contains the four spectral bands of each of the 9 pixels in the 3×3 neighborhood and the class of the central pixel which was one of the six classes: red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble, very damp grey soil. The examples were randomized. The data were divided into a training set and a test set with 4,435 examples in the training set and 2,000 in the test set. This data set was obtained from STATLOG project [13].

4.5 Letter Data Set

This dataset is a well-known benchmark, constructed by David J. Slate of Odesta Corporation, Evanston, IL 6020. The objective here is to classify each of a large number of black and white rectangular pixel displays as one of the 26 capital letters of the English alphabet. The character images produced were based on 20 different fonts and each letter within these fonts was randomly distorted to produce a file of 20,000 unique images. For each image, 16 numerical attributes were calculated using edge counts and measures of statistical moments. The attributes were scaled and discretized into a range of integer values from 0 to 15. The size of the training set is the first 15000 items and the resulting model is used to predict the letter category for the remaining 5000. This data set is one of the data sets used in the STATLOG project [13].

Table 1 and Table 2 show the CPU elapsed time in training and testing respectively. Table 3 and Table 4 show the number of prototypes/hyperspheres and the test classification error (%) respectively. As can be seen from the results, although SCHS requires considerably large training time compared to the other methods, it generates the smallest number of prototypes and thus has the fastest classification time. Moreover, it has the minimum test classification error for most data sets.

Table 1. Comparison of CPU elapsed time in training

Data Set	LVQ	RCE	RCE-2	SCHS
I-I	0.39	4.59	3.34	16.13
Parabola	0.31	1.67	1.67	4.7
Thyroid	25.76	193.94	75.02	134.44
Satimage	31.58	1430.01	98.94	113.84
Letter	268.29	1071.00	641.36	525.75

Table 2. Comparison of CPU elapsed time in testing

Data Set	LVQ	RCE	RCE-2	SCHS
I-I	0.86	2.95	3.28	0.09
Parabola	0.80	1.11	1.72	0.11
Thyroid	1.34	1.69	1.48	0.06
Satimage	1.43	1.69	1.66	1.05
Letter	9.53	9.73	9.00	5.13

Table 3. Comparison of number of prototypes / hyperspheres

Data Set	LVQ	RCE	RCE-2	SCHS
I-I	161	274	307	11
Parabola	161	28	78	12
Thyroid	2161	1133	1028	42
Satimage	1777	924	938	618
Letter	6006	3009	2914	2010

Table 4. Comparison of test classification error (%)

Data Set	LVQ	RCE	RCE-2	SCHS
I-I	13.90	25.44	17.09	11.54
Parabola	4.03	15.49	3.90	3.23
Thyroid	7.44	17.78	9.72	7.28
Satimage	11.85	21.10	12.55	9.05
Letter	6.78	18.66	8.18	7.24

5 Conclusions

In this article, we presented a novel method (SCHS) for clustering class regions using hyperspheres. This method has some distinct that do not exist in RCE methods. 1) Positions of hyperspheres' centers and their radii do not depend on the order of presentation of training examples to the network. 2) Storage requirements and number of training epochs are also not affected by the order of presentation of training examples to the network. 3) Hyperspheres' centers are not restricted to be a subset of the training set. Conversely, they are learned via an optimization procedure. 4) Hyperspheres can be permitted to enclose patterns of other classes and hence it has better generalization capability (especially for noisy problems). Our experiments show that the proposed method needs small storage compared to LVQ, RCE and RCE-2 methods and hence achieves a significant acceleration in the classification computation. Moreover, in most data sets, it has better performance than the other methods.

References

1. Hastie, T., Tibshirani, R.: Discriminant adaptive nearest neighbor classification. IEEE Trans. Pattern Analysis and Machine Intelligence. 18(6) (1996) 607-616.
2. Reilly, D., Cooper, L., Elbaum, C.: A neural model for category learning. Biological Cybernetics. 45 (1982) 35-41.
3. Hudak, M.J.: RCE Classifiers: Theory and Practice. Cybernetics and Systems: An International Journal. 23 (1992) 483-515.
4. Tsumura, N.,Itoh, K., Ichioka, Y.: Reliable classification by double hyperspheres in pattern vector space, Pattern Recognition. 28 (1995) 1621-1626.

5. Kositsky, M., Ullman, S.: Learning Class Regions by the Union of Ellipsoids. In Proc. 13th International Conference on Pattern Recognition (ICPR) 4, IEEE Computer Society Press (1996) 750-757.
6. Atiya, A., Hashem, S., Fayed, H.: New hyperspheres for pattern classification. In Proc. 1st International Computer Engineering Conference (ICENCO-2004), Cairo, Egypt (2004) 258–263.
7. Wright, S.J.: Primal-Dual Interior-Point Methods. Philadelphia: SIAM Publications, (1997).
8. Khachiyan, L.G.: Rounding of polytopes in the real number model of computation. Mathematics of Operations Research. 21 (1996) 307-320.
9. Kohonen, T.: Self-organization and Associative Memory. 3rd edn. Springer-Verlag, Heidelberg, Germany (1989).
10. Hastie, T., Tisbshirani, R., Friedman, J.: The Elements of Statistical Learning, Springer (2001).
11. Zhang, H., Sun, G.: Optimal reference subset selection for nearest neighbor classification by tabu search. Pattern Recognition. 35 (2002) 1481-1490.
12. Prechelt, L.: Proben1. A Set of Neural-Network Benchmark Problems. Univ. Karlsruhe, Germany. Available FTP: ira.uka.de/pub/neuron/proben1.tar.gz (1994).
13. Michie, D., Spiegelhalter, D.J., Taylor, C.C.: Machine Learning, Neural and Statistical Classification. New-York, NY: Ellis Horwood (1994).

Direct Estimation of Fault Tolerance of Feedforward Neural Networks in Pattern Recognition

Huilan Jiang, Tangsheng Liu, and Mengbin Wang

Tianjin University, Tianjin 300072, China
{Huilan Jiang, Tangsheng Liu, Mengbin Wang}hljiang65@126.com

Abstract. This paper studies fault-tolerance problem of feedforward neural networks implemented in pattern recognition. Based on dynamical system theory, two concepts of pseudo-attractor and its region of attraction are introduced. A method estimating fault tolerance of feedforward neural networks has been developed. This paper also presents definitions of terminologies and detailed derivations of the methodology. Some preliminary results of case studies using the proposed method are shown, the proposed method has provided a framework and an efficient way for direct evaluation of fault-tolerance in feedforward neural networks.

1 Introduction

The feedforward neural network (FFNN) is the most popular NN in both academic research and practical applications. The FFNN has been applied in many engineering areas due to its outstanding capabilities of non-linear function approximation, classification, and parallel processing[1]. Especially, it has been implemented in real-time information systems dealing with the pattern recognition (PR) problems.

To understand NNs, one should study relevant topics of network structure, learning algorithm, memory capacity, generalization ability as well as fault tolerance. Current researches in FFNN have been focusing on network structure, learning algorithm and convergence speed. In view of engineering applications, the generalization ability of FFNN is, however, of the highest importance. When implemented in the real-time information processing systems, the input to FFNN is directly acquired on-site where data is possibly polluted with noises or transmission errors, this will result in uncertainties of the system output, consequently the feasibility of such system can not be evaluated precisely. It is therefore impractical to evaluate the NN-based real-time information systems without studying the fault-tolerance of NNs.

So far, there are quite a few research works that have been carried out regarding generalization and fault-tolerance of FFNNs. However, most of these research works have concentrated on the influences from the network structure and training data distribution. In [2] Hao et al adjust the FFNN's performance by varying its network structure. Dynamic pruning technique has been employed to simplify the NN. In [3] and [4], Initial conditions and distribution of training samples have been studied to improve the network performance. Based on statistics and independent component

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 124 – 131, 2006.

analysis, [5] and [6] select the training samples with dominant features. The influence from training samples quantity and quality on network performance is also studied. [7] utilizes Genetic Algorithms (GAs) to obtain the optimal structure for NNs. Nevertheless, it fails to provide an explicit analytical approach for fault tolerance measurement in FFNNs. [8] has made some important progresses. It has proposed to measure fault-tolerance using the attraction regions of stable attractors, based on dynamical system theory. However, unlike the recurrent networks the FFNN is not a dynamical system as the inputs simply propagates forward on a layer-to-layer basis. As such, the attractor and its region of attraction do not exist in FFNNs.

This paper introduce a concept of pseudo-attractor in FFNNs. A direct method using the attraction region of pseudo-attractor for fault-tolerance evaluation is also developed and detailed mathematical derivations have been presented. It has provided a practical way and a framework for measuring fault-tolerance of FFNNs in PR.

2 Architecture of FFNN

FFNN is a distributed processing system with capability of nonlinear approximation. The overall structure of a 3-layer FFNN is shown in Fig. 1. The neurons of adjacent layers are fully connected by synaptic weights. The input propagates forward through the network, and the final output is obtained at the output layer.

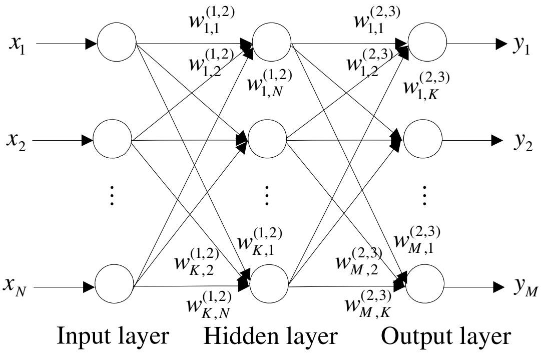


Fig. 1. Architecture graph of feedforward neural network

The FFNN is trained in a supervised learning manner, which employs the error back-propagation algorithm. There are several steps involved in the algorithm:

1) Initialization. Set all the synaptic weights and thresholds with random numbers;

2) Presentation of training samples. Input training data into the network.

3) Forward computation. Based on the input, weights and thresholds, calculate every neuron's output. Training is considered to be terminated when the sum of the squared error (between desired and calculated output) per epoch is sufficiently small. Otherwise, the training will go to step *4)*.

4) Backward computation. Adjust the weights layer by layer in a backward direction, then switch to step *2).*

3 State of Art of Fault Tolerance in Neural Networks

The fault tolerance is one of most important research topics in NNs, especially for the networks using to practical project. The existing methods usually describe the fault-tolerance in terms of several indexes, e.g. the volume of attraction region [8]. A NN can be viewed as a dynamical system. Therefore, the stability of the network can be described in terms of the attractor and corresponding region of attraction.

Definition: Attractor
For a given network, if a vector $\boldsymbol{X}$ satisfies $\boldsymbol{X} = sgn(\boldsymbol{WX}\text{-}\boldsymbol{\theta})$ (where $\boldsymbol{W}$ and $\boldsymbol{\theta}$ are the weight and threshold vectors of the network respectively), then $\boldsymbol{X}$ is called the attractor of the system.

Definition: Region of attraction (Attractor)
If $\boldsymbol{Y}$ is an attractor, and if there exists a trajectory from $\boldsymbol{X}$ to $\boldsymbol{Y}$, then $\boldsymbol{X}$ is said to be attracted to $\boldsymbol{Y}$, noted as $\boldsymbol{X} \xrightarrow{W} \boldsymbol{Y}$. If all $\boldsymbol{X} \in N(\boldsymbol{Y})$ stratify $\boldsymbol{X} \xrightarrow{W} \boldsymbol{Y}$, then $\boldsymbol{N(Y)}$ is the region of attraction of $\boldsymbol{Y}$**.**

For stable attractors, the volume of attraction region can be used to estimate the fault-tolerance of NNs. The volume is defined as the total number of input states within the region of attraction, denoted as B_α , $\alpha = 1,2,\cdots,M$. Apparently, larger B_α indicates better attraction capability of attractor α . However, this index may give illusive information about fault-tolerance. For example, if we add *n* redundant input nodes in the input layer of a NN and their weights connected to neurons are set all zeros, the samples' region of attraction will increase 2^n times accordingly. By doing so, the regions of attraction of those samples are not really increased. Therefore, the ratio of $R_\alpha = B_\alpha /(\Omega_0 / M)$, which can be termed as normalized attraction region, is more accurate for fault-tolerance estimation; where $\Omega_0 = 2^N$ is the total volume of the input space given the network is *N* dimensional and the input (output) is binary data. Subsequently, the ratio of (1), which is the average of normalized attraction regions, reflects overall fault tolerance of the entire network and the fault tolerance of the NN.

$$R_s = \sum_{\alpha=1}^{M} R_\alpha / M = \sum_{\alpha=1}^{M} {}^{B_\alpha}\!/\!_{\Omega_0} \qquad (1)$$

4 Direct Estimation of Fault Tolerance in FFNN

Although indexes like B_α and R_s have been defined and can be used to evaluate fault tolerance, methods for direct calculation of these indexes are not available. Current methods examine convergence of every testing sample by global searching method. For any networks of higher dimensionalities, it is impractical to use global searching approach. It is therefore necessary to develop a new method to calculate the fault-tolerance directly. Based on the dynamical system theory, this paper proposes a direct method for evaluating fault tolerance in FFNN. The new method is based on pseudo-attractor and its region of attraction, which will be introduced in the following sections.

A Pseudo-attractor and Its Region of Attraction

Mathematically speaking, the feedforward network is actually a mapping from the input space to the output space. Subsequently, pseudo-attractor and its region of attraction are defined as below.

Definition: Pseudo-attractor

For a FFNN network, the inputs and outputs (targets) of samples memorized after training form a pair of training samples, i.e. $(\boldsymbol{X}_k, \boldsymbol{Y}_k)$, k = 1, 2, … M. The stable state (or output vector) $\boldsymbol{Y}_k$ is defined as the pseudo-attractor of FFNN.

Definition: Region of attraction (Pseudo-attractor)

Given a FFNN network: $\boldsymbol{F}:\boldsymbol{A}\rightarrow\boldsymbol{B}$, the counterparts corresponding to pseudo-attractor $\boldsymbol{Y}_k$ in the input set **A**, $\boldsymbol{F}^{-1}(\boldsymbol{Y}_k)=\{\boldsymbol{X}:\boldsymbol{F}(\boldsymbol{X})=\boldsymbol{Y}_k\}$ form the region of attraction of $\boldsymbol{Y}_k$.

Fig.2 gives a graphical illustration of the pseudo attractor and its region of attraction.

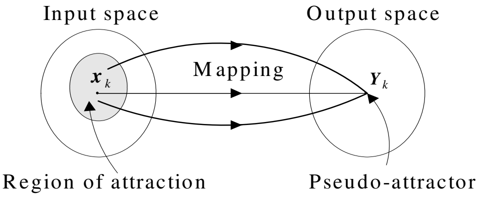


Fig. 2. Illustration of pseudo-attractor and its region of attraction

B Direct Calculation of Region of Attraction of a FFNN

With the new concepts defined above, a direct method for fault tolerance estimation in feedforward networks is proposed. To apply the new method, we should first determine the pseudo-attractors among the output set of the network. Next, the attraction region of the pseudo-attractor can be obtained. Subsequently, the indices of R_s will be calculated to estimate fault tolerance of NNs. Since a FFNN network usually consists of several distinct layers, the calculation of the attraction region should begin with the network of single layer.

1) Attraction region of single layer networks

Each layer of the network in Fig 1 represents a single layer network. To decide if output vector $\boldsymbol{Y}=(y_1, y_2, \cdots, y_M)^{-1}$ is the pseudo-attractor and to calculate its region of attraction, the linear equations in (2) should be resolved,

$$\begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1n} \\ w_{20} & w_{21} & \cdots & w_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ w_{mo} & w_{m1} & \cdots & w_{mn} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = f^{-1} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \tag{2}$$

or $\boldsymbol{WX} = f^{-1}(\boldsymbol{Y})$, where f^{-1} is the inverse function of the activation function *f*, where $w_{j0}= \theta_j$, j=1,2,…,*M*, and $x_0 = -1$.

Should linear equation group (2) be solvable, ***Y*** is the pseudo-attractor of the NN, and the set of solutions is the attraction region of ***Y*** (x_0 is not included).

2) Attraction region of a three layer network with any activation function f

Consider a three layers FFNN with *N*-dimensional input $\boldsymbol{X}=(x_1,x_2,\ldots,x_N)^{-1}\in\boldsymbol{R}^N$. Suppose there are *K* and *M* neurons in the hidden and output layer, then the output of each layer will be $\boldsymbol{H}=(h_1,h_2,\ldots,h_k)^{-1}\in\boldsymbol{R}^K$ and $\boldsymbol{Y}=(y_1,y_2,\ldots,y_M)^{-1}\in\boldsymbol{R}^M$ respectively. The weights between different layers are $\boldsymbol{W}^{(1,2)} = (w_{k,i}^{(1,2)})_{K\times(N+1)}$ and $\boldsymbol{W}^{(2,3)} = (w_{m,k})_{M\times(K+1)}$, where *i*=0,1,2,…,*N*, *k*=1,2,…,*K*, *m*=1,2,…,*M*; $w_{k0} = \theta_k^{(1)}$, and $w_{m0} = \theta_m^{(2)}$. Consequently the relationship between ***X*** and ***Y*** can be expressed as:

$$\begin{aligned} \boldsymbol{Y} &= f(\boldsymbol{W}^{(2,3)}\boldsymbol{H}) \\ \boldsymbol{H} &= f(\boldsymbol{W}^{(1,2)}\boldsymbol{X}) \end{aligned} \tag{3}$$

For a given vector ***Y***, should the following equations in (4) be solvable, ***Y*** is the pseudo-attractor and the solution set of ***X*** is the region of attraction.

$$\begin{aligned} f^{-1}(\boldsymbol{Y}) &= \boldsymbol{W}^{(2,3)}\boldsymbol{H} \\ f^{-1}(\boldsymbol{H}) &= \boldsymbol{W}^{(1,2)}\boldsymbol{X} \end{aligned} \tag{4}$$

3) The way of solving equation

In brief, a NN must be trained using relevant algorithm before to estimate its fault tolerance using the method of pseudo-attractor. The trained weight and threshold vectors ***W*** are then utilized to solve the equations in (4) to calculate the volume of the region of attraction.

Equation (4) is a non-homogeneous linearly equation group. Detail method of solving non-homogeneous linearly equation may refer to foundation tutorial of engineering mathematics [9]. Here only gives basic solving steps.

Step1: For equation $f^{-1}(\boldsymbol{Y}) = \boldsymbol{W}^{(2,3)}\boldsymbol{H}$, solving its specific solution ξ_*' and basic solution series $\boldsymbol{\eta}_1', \boldsymbol{\eta}_2', \cdots, \boldsymbol{\eta}_n'$;

Step2: Utilizing the result of step1 to solve specific solution ξ_* and basic solution series $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \cdots, \boldsymbol{\eta}_n$ of equation $f^{-1}(\boldsymbol{H}) = \boldsymbol{W}^{(1,2)}\boldsymbol{X}$ and obtain final solution $\boldsymbol{X} = \xi_* + k_1\boldsymbol{\eta}_1 + k_2\boldsymbol{\eta}_2 + \cdots + k_n\boldsymbol{\eta}_n$ of corresponding to attractor $\boldsymbol{Y}$;

Step3: For all detecting samples X^i (i=1,2,…,S) organized from training sample in input state space, to test in turn if they fall in region of attraction of $\boldsymbol{Y}$ or not.

If $\boldsymbol{X}^i = \xi_* + k_1\boldsymbol{\eta}_1 + k_2\boldsymbol{\eta}_2 + \cdots + k_n\boldsymbol{\eta}_n$ has unique solution of $k_1, k_2, \cdots, k_n$, $\boldsymbol{X}^i$ is in the region of attraction of $\boldsymbol{Y}$;

Step4: After testing all detecting samples, R_S that indicates the index of region of attraction is obtained.

5 Simulation Result

There is no limit for memory volume in the FFNNs. The training samples will be eventually memorized as stable pseudo-attractors provided training has appropriately converged as desired. However, the corresponding attraction regions are different for different kinds of structure of NNs. In our experimentation, two FFNNs have been implemented for the binary PR to investigate fault tolerance capability in FFNN.

Two FFNN networks that this paper selected to do analysis are respectively: the one is typical BPNN with sigmond activation function expressed as formula (5), and its model is the structure of 8-16-10; the other one is RBFNN with gauss kernel function expressed as formula (6), and its model is the structure of 8-10-10.

$$\begin{cases} I = \boldsymbol{W}\boldsymbol{X}^T - \theta = \sum_{i=1}^{N} w_{ji}x_i - \theta \\ y = \dfrac{1}{1+\exp(-\beta I)} \end{cases} \tag{5}$$

$$y_j = \exp(\frac{(\boldsymbol{X}-z_j)^T(\boldsymbol{X}-z_j)}{2\sigma^2}) \tag{6}$$

where y_j is the output of j th node; $\boldsymbol{X}$=(x_1, x_2,…, x_n)$^{\mathrm{T}}$ denotes input vector;

z_j denotes the central value of gauss function; σ is standard constant.

In each case, the NN has been trained using a ten binary sample data set, then each network is tested to recognize 80 detecting samples based on their training samples. To simulate real-world environment where data is often polluted, the testing samples differ slightly from the training samples. The deviation is kept in only one bit, while more serious errors could be considered without fundamental difference for our study.

The simulation results are given in Table 1. The R_S respectively in two cases indicate that the FFNN in case one has better capability of fault-tolerance than in case two. The influence factor is the sample variances which measure the sample distribution diversity. As observed, the same variance of $\sqrt{2}$ is obtained in case one while various variances are obtained in case two. This indicates that the training samples of case one are distributed more evenly comparing to that of case two, which results in a higher R_S of the FFNN in case one. Thus it can be concluded that the distribution of training samples (input) influences the fault-tolerance of FFNN and better distribution implies higher fault-tolerance – see Appendix for more details. On the other hand, from comparing results of two NN structures, it can be shown that fault-tolerance ability of

RBFNN is higher than that of BPNN. This is because RBFNN has special local response characteristic and is able to attract strongly input of detecting sample to corresponding pseudo-attractors.

Table 1. Simulation results of two FFNNs in two cases

Training samples of Case one			BPNN		RBFNN	
Index	Input	Output	B_α	R_s	B_α	R_s
1	00000000	1000000000	1	31.3%	8	100%
2	00001111	0100000000	2		8	
3	11110000	0010000000	4		8	
4	11001100	0001000000	4		8	
5	00110011	0000100000	2		8	
6	10101010	0000010000	1		8	
7	01010101	0000001000	3		8	
8	11000011	0000000100	1		8	
9	00111100	0000000010	2		8	
10	11111111	0000000001	5		8	
Sample variance	$\sqrt{2}$, $\sqrt{2}$, $\sqrt{2}$, $\sqrt{2}$, $\sqrt{2}$, $\sqrt{2}$, $\sqrt{2}$, $\sqrt{2}$, $\sqrt{2}$, $\sqrt{2}$					
Training samples of Case two			BPNN		RBFNN	
Index	Input	Output	B_α	R_s	B_α	R_s
1	00100000	1000000000	0	11.3%	8	92.5%
2	00001101	0100000000	1		8	
3	11110100	0010000000	1		7	
4	11010100	0001000000	2		7	
5	00110011	0000100000	0		7	
6	10100011	0000010000	1		7	
7	01010101	0000001000	1		7	
8	11000011	0000000100	0		8	
9	00110100	0000000010	0		7	
10	10111111	0000000001	3		8	
Sample variance	1.36, 1.50, 1.28, 1.36, 1.28, 1.36, 1.28, 1.50 ,1.20 ,1.43					

6 Future Scope and Conclusion

Fault-tolerance capability is essential for NNs in various engineering applications such as the PR. It is therefore imperative to carry out theoretical research of fault-tolerance in FFNNs, which have been most widely used. In this paper, we have proposed a new method to evaluate the fault-tolerance based on pseudo-attractor. The proposed method is able to estimate directly the fault-tolerance without excessive efforts by traditional global searching or experimental methods. Simulation with pattern recognition tasks has been carried out with the proposed method in the paper. Future work is underway towards enhancing fault-tolerance in feedforward network through different methods of training.

References

1. S. Haykin, Neural Network-A comprehensive Foundation. Prentice -Hall Inc , 1994
2. P. Hao, W. Xiao, et al. “Investigation of structure Variation of BP Network”, Control and Decision, vol. 16, pp. 287-298, 2001.
3. A. Atiya, and C. Ji, “How initial conditions affect generalization performance in large networks”, IEEE Trans. on Neural Networks, vol. 8, pp. 448-451, 1997.
4. N. Kwak, and C. H. Choi, “Input feature selection for classification problems”, IEEE Trans. on Neural Networks, vol. 13, pp. 143-159, 2002.
5. A. D. Back, T. P. Trappenberg, “Selecting inputs for modeling using normalized higher order statistics and independent component analysis”, IEEE Trans. on Neural Networks, vol. 12, pp. 612-617, 2001.
6. D. Y. Ywung, “Constructive neural network as estimators of bayesian discriminant Func tion”, Pattern Recognition, vol. 26, pp. 189-204, 1993.
7. M. Mcinerney, A. P. Dhawan, “Use of genetic algorithm with back propagation in train ing of feedforward neural networks (Published Conference Proceedings style)”, 1993 IEEE Int. Conf. Neural Networks, San Francisco, 1993, pp. 203-208.
8. F. Zhang, and G. Zhao, “Some issues about neural networks of associative memory”, Automation, vol. 20, pp. 513-521, 1994.
9. J. L. Liu, Z. S. Yang, and S. B. Zeng, “Foundation tutorial of engineering mathematics (Book style)”, Tianjin University Press, 2000.

Appendix

The paper uses distance among sample data points to represent the degree of the distribution diversity. This is only one of the approaches to measure sample distribution diversity. Based on this approach, the distance among sample points can be expressed as,

$$d(\boldsymbol{x},\boldsymbol{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$

where $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ and $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)$.

For the simulation presented in the paper, the distribution diversity can be measured by the distance defined above. Specifically for each simulation case, the set of input sample data is $\{\boldsymbol{x}^{(i)}\}$, $i = 1,2,\ldots,m$, i.e. a total of m samples have been used. The i-th sample is $\boldsymbol{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \cdots x_n^{(i)})$. The centre of this set is identified as

$$z = (z_1, z_2, \cdots, z_n) \ : \ z_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)} \text{ , where } j = 1, 2, \ldots, n$$

The distribution diversity of input samples in the input space can be expressed by the relative distance to this centre point. If the all sample points have the same distance to the centre then the sample data are said to be evenly distributed in the space. Otherwise, they are not evenly distributed.

A Fully Automated Pattern Classification Method of Combining Self-Organizing Map with Generalization Regression Neural Network

Chao-feng Li[1], Jun-ben Zhang[1], Zheng-you Wang[2], and Shi-tong Wang[1]

[1] School of Information Technology, Southern Yangtze University, Wuxi 214122, China
chaofeng.li@163.com
[2] School of Information Technology, Jiangxi Univ. of Finance & Economics, Nanchang 330013, China

Abstract. The paper presents a new automated pattern classification method. At first original data points are partitioned by unsupervised self-organizing map network (SOM). Then from the above clustering results, some labelled points nearer to each clustering center are chosen to train supervised generalization regression neural network model (GRNN). Then utilizing the decided GRNN model, we reclassify these original data points and gain new clustering results. At last from new clustering results, we choose some labelled points nearer to new clustering center to train and classify again, and so repeat until clustering center no longer changes. Experimental results for Iris data, Wine data and remote sensing data verify the validity of our method.

1 Introduction

Artificial neural networks have been employed for many years in pattern recognition [1,2]. In general, these models are composed of many nonlinear computational elements (neural nodes) operating in parallel and arranged in patterns reminiscent of biological neural nets. Similar to pattern recognition, there exist two types of modes for neural networks – unsupervised and supervised. The unsupervised type of these networks, which possesses the self-organizing property, is called competitive learning networks [2], for example SOM. It doesn't require human to have the foreknowledge of the classes, and mainly uses some clustering algorithm to classify original data [3], but it usually gains baddish classification results. The supervised method has usually better classification effects and is most commonly adopted in factual application, but it needs many appropriate labeling training samples that are sometimes difficultly gained. That is to say, the unsupervised and supervised methods have each advantages and limitation.

In order to integrate their advantages of unsupervised and supervised methods and realize automated classification with high quality, the paper presents a hybrid classification method of combing unsupervised SOM network with supervised GRNN. It firstly uses SOM to partition original data points, and then from the clustering results chooses some labelled training samples for GRNN to train and reclassify, and so repeat to gain best classification results.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 132 – 139, 2006.

This paper is organized as follows. Section 2 briefly introduces the basic theory of SOM. In Section 3, the principle and structure of GRNN is described. Section 4 gives a hybrid classification algorithm of combing SOM and GRNN. Several experimental results comparison and analysis are done in Section 5. Finally, we draw some simple conclusions in Section 6.

2 Brief Introduction to Self-Organizing Map

SOM network is a two layers network proposed by T. Kohonen in 1981 [4]. The first layer is input layer, which consists of a sample of n-dimensional data vectors, namely

$$x(t) = \left[x_1(t), x_2(t), \cdots, x_n(t)\right] \tag{1}$$

where t is regarded as the index of the data vectors in the sample and also the index of the iterations (t=1, 2, …, T), and n is the number of dimensions or features.

The second layer is output layer, and its output nodes usually be arranged in the form of two-dimension array. Every input node is entirely connected with output node by dynamic weights vector, and the connection weight vector is:

$$w_i(t) = \left[w_{i1}(t), w_{i2}(t), \cdots, w_{in}(t)\right] \tag{2}$$

where i denotes the index of the neuron in the SOM (i=1,2,…, I). I, the number of nodes, is determined empirically.

In network, every output node has a topology neighbor, and the size of neighbor is changing with the training process. For each intermediate iteration t, the training process performs the following steps:

The best matching neuron $w_c(t)$ most closely resembling the current data vector c(t) is selected, for which the following is true:

$$\|x(t) - w_c(t)\| = \min_i \left\{\|x(t) - w_i(t)\|\right\} \tag{3}$$

The nodes w_i are updated, using the formula:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)[x_i(t) - w_{ij}(t)] \tag{4}$$

where the adjustment is monotonically decreasing with the number of iterations. This is controlled by the learning rate factor $\eta(t)$ (0<$\eta(t)$<1), which is usually defined as a linearly decreasing function over the iterations.

3 Introduction to Generalized Regression Neural Network

Generalized Regression Neural Network is a new type of Neural Network proposed by Donald F. Specht in 1991. Compared to the BPNN, GRNN has a lot of advantages [5,6], namely

(1) The weights of each layer and the number of hidden layer nodes can be decided only by the training samples.

(2) It needn't iteration during training process.

(3) When network-operating mode changes, only that needed to modify the corresponding training samples and reconstruct network.

The GRNN is used for estimation of continuous variables, as in standard regression techniques. It is related to the radial basis function network and is based on established statistical principles and converges with an increasing number of samples asymptotically to the optimal regression surface.

Suppose the vector x and scalar y are random variants, X and Y are observation values, and f (x, y) is defined a joint continuous probability density function. If the f(x, y) is known, then the regression of y on x is given [7] by

$$E\left[y|x\right]=\frac{\int_{-\infty}^{\infty} yf(x,y)dy}{\int_{-\infty}^{\infty} f(x,y)dy} \tag{5}$$

When the density f(x, y) is not known, it must usually be estimated from a sample of observations of x and y. The probability estimator $\overset{\Lambda}{f}(x,y)$ is based upon sample values x_i and y_i of the random variables x and y.

$$\overset{\Lambda}{f}(x,y)=\frac{1}{(2\pi)^{(m+1)/2}\sigma^{(m+1)}}\cdot\frac{1}{n}\cdot\sum_{i=1}^{n}\exp\left(-\frac{(x-x_i)^{\mathrm{T}}(x-x_i)}{2\sigma^2}\right)\exp\left(-\frac{(y-y_i)^2}{2\sigma^2}\right) \tag{6}$$

where m is the dimension of the vector variable x, and n is the number of sample observations, and σ is the spread parameter.

Defining the scalar function D_i^2

$$D_i^2=(x-x_i)^{\mathrm{T}}(x-x_i) \tag{7}$$

And performing the indicated integrations yields the following:

$$\overset{\Lambda}{y}(x)=\frac{\sum_{i=1}^{n} y_i\exp(-D_i^2/2\sigma^2)}{\sum_{i=1}^{n}\exp(-D_i^2/2\sigma^2)} \tag{8}$$

Schematic diagram of GRNN architecture is presented in Fig.1. Differing from the LMBPN, the GRNN consists of four layers: input layer, pattern layer, summation layer and output layer. The input layer has m units and receives the input vector. The pattern layer has n units, which calculates and outputs the value of kernel function

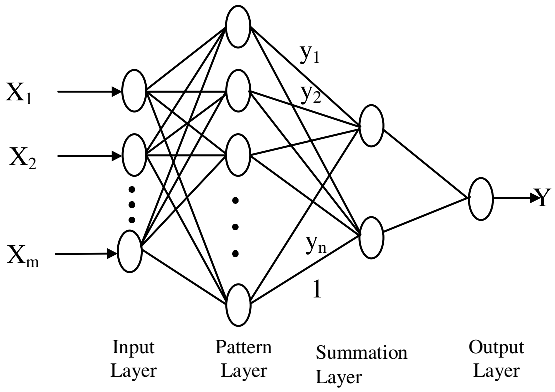


Fig. 1. Schematic diagram of GRNN architecture

$\exp\left(-\frac{D_i^2}{2\sigma^2}\right)$. The summation layer has two units, and its output is the value of $\sum_{i=1}^{n} y_i \exp(-D_i^2/2\sigma^2)$ and $\sum_{i=1}^{n} \exp(-D_i^2/2\sigma^2)$. The output layer has one unit and its output is the value of $\overset{\Lambda}{y}(x)$.

4 A Hybrid Classification Algorithm of Combining SOM and GRNN

The paper presents a fully automated classification method of combing unsupervised SOM and supervised GRNN, and the whole algorithm is as follows.

Step 1: Suppose $X = [x_1, x_2, ..., x_n]^{\mathrm{T}}$ is input samples, and then X is normalized by equation $X = \frac{X}{\|X\|}$, $\|X\|$ is its norm.

Step 2: Use SOM network to cluster original data points and gain each class center and labelled samples.

Step 3: Calculate distance d_{ij} between each data point and class center, and then choose some labelled samples that are nearer to class center.

Step 4: Make advantage of gained labelled samples to train GRNN model, and then reclassify original data points to gain new-labelled samples.

Step 5: Update class center. According to classification results by GRNN, we gain new class center, and if the distance between new clustering center and former class center is less than a given tiny threshold value, the classification procedure is stopped, else going to (3) to continue.

5 Experimental Results Comparison and Analysis

5.1 Experimental Results and Comparison for Iris Data Classification

Iris data set is with 150 random samples of flowers from the iris species Setosa, Versicolor, and Virginica collected by Anderson (1935), and contains 3 classes of 50 instances each. One class is linearly separable from the other two classes, and the latter are not linearly separable from each other.

We use SOM to partition and gain classification results shown in table 1, and then use the hybrid classification algorithm to gain classification results shown in table 2. For further comparison we randomly choose 20 samples for training from each class, the other 30 for testing, and use single supervised GRNN to classify and gain classification results shown in table 3. (The value of parameter σ in above two GRNN is 0.05).

Table 1. Iris data classification results of SOM

Class	Setosa	Versicolor	Virginica	Accuracy	Average Accuracy
Setosa	50	0	0	100%	90.7%
Versicolor	0	36	14	72%	
Virginica	0	0	50	100%	

Table 2. Iris data classification results of the hybrid classification algorithm

Class	Setosa	Versicolor	Virginica	Accuracy (%)	Average Accuracy (%)
Setosa	50	0	0	100	97.3
Versicolor	0	48	2	96	
Virginica	0	2	48	96	

Table 3. Iris data classification results of single GRNN

Class	Setosa	Versicolor	Virginica	Accuracy (%)	Average Accuracy (%)
Setosa	30	0	0	100	97.5
Versicolor	0	0.75	29.25	97.5	
Virginica	0	1.5	28.5	95	

From above table 1, table 2 and table 3, we can find the hybrid classification algorithm is far superior to single unsupervised SOM classifier and improves about 6.6% in accuracy, and it is a little inferior to single supervised GRNN classifier, but it needn't choose training sample by manual and is a fully automated classification method.

5.2 Experimental Results and Comparison for Wine Data Classification

Wine data set is the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars, and usually used for comparing various classifiers. It has three types, and here we respectively marked as class A, class B, class C, and class A has 59 samples, class B 71 samples, class C 48 samples.

We use SOM to partition and gain classification results shown in table 4, and then use the hybrid classification algorithm to gain classification results shown in table 5 (where the value of parameter σ in GRNN is 0.05). For further comparison we randomly choose half samples for training from each class, and the other half for testing, and use single supervised GRNN to classify and gain classification results shown in table 6 (where the value of parameter σ in GRNN is also 0.05).

Table 4. Wine data classification results of SOM

Class	A	B	C	Accuracy (%)	Average Accuracy (%)
A	58	1	0	98.30	82.0
B	16	40	15	56.34	
C	0	0	48	100	

Table 5. Wine data classification results of the hybrid classification algorithm

Class	A	B	C	Accuracy (%)	Average Accuracy (%)
A	59	0	0	100	94.9
B	4	62	5	87.3	
C	0	0	48	100	

Table 6. Wine data classification results of single GRNN

Class	A	B	C	Accuracy (%)	Average Accuracy (%)
A	28	1	0	96.6	93.3
B	4	31	1	86.1	
C	0	0	24	100	

From above table 4, table 5 and table 6, we can find the hybrid classification algorithm is far superior to single unsupervised SOM classifier and improves about 12.9% in accuracy. Moreover the hybrid classifier is a little superior to single supervised GRNN classifier and improves about 1.6%, which is possibly as a result of gaining better training samples by SOM clustering algorithm.

5.3 Experimental Results and Comparison for Remote Sensing Image

An experimental remote sensing data sampled from satellite TM image. According to the terrain map, analyzing the image visually, we divide it into 6 categories, namely road, city area, field, green-land, hill, water, and manually choose samples data set for six categories and gain 1200 samples for experiment (200 for each category).

We use SOM model to gain clustering results shown in table 7, and then use the hybrid classification algorithm to gain classification results shown in table 8 (here the value of parameter σ in GRNN is 0.05). For further comparison we randomly choose 50 samples for training from each class, the other 150 for testing, and use single supervised GRNN to classify and gain classification results shown in table 9 (here the value of parameter σ in GRNN is also 0.05).

Table 7. Remote Sensing data classification results of SOM

Class	Road	City	Field	Green	Hill	Water	Accuracy (%)	Average Accuracy
Road	193	7	0	0	0	0	96.5	67.3
City	111	0	0	0	89	0	0	
Field	0	0	48	152	0	0	24.0	
Green	1	0	2	197	0	0	98.5	
Hill	0	0	1	0	182	17	91.0	
Water	0	0	0	0	12	188	94.0	

Table 8. Remote Sensing data classification results of the hybrid classification algorithm

Class	Road	City	Field	Green	Hill	Water	Accuracy (%)	Average Accuracy
Road	191	9	0	0	0	0	95.5	95.8
City	21	176	0	1	2	0	88.0	
Field	0	0	192	8	0	0	96.0	
Green	0	1	2	197	0	0	98.5	
Hill	0	1	0	0	199	0	99.5	
Water	0	1	0	0	5	194	97.0	

Table 9. Remote Sensing data classification results of GRNN

Class	Road	City	Field	Green	Hill	Water	Accuracy (%)	Average Accuracy
Road	149	1	0	0	0	0	99.3	94.0
City	24	125	1	0	0	0	83.3	
Field	0	0	140	10	0	0	93.3	
Green	0	0	0	150	0	0	100	
Hill	0	1	1	1	139	8	92.7	
Water	0	2	0	0	5	143	95.3	

From above table 7, table 8 and table 9, we can find the hybrid classification algorithm is far superior to single unsupervised SOM classifier and improves about 28.5% in accuracy. Moreover the hybrid classifier is a little superior to single supervised GRNN classifier and improves about 1.8%.

All above three kinds of data experimental results show the validity of our proposed hybrid classification algorithm.

6 Conclusions

Aiming at each limitation of supervised and unsupervised classification method, the paper presents a fully automated classification method of combing unsupervised SOM and supervised GRNN. The hybrid classifier firstly uses SOM to partition original data points, and then from clustering results chooses labelled training samples for GRNN to train and reclassify. Experimental results for Iris data, Wine data and remote sensing data show our method is absolutely effective.

References

1. Bishop C.M.: Neural Networks for Pattern Recognition. Oxford (1995)
2. Hertz J., Krogh A., Palmer R.: Introduction to the Theory of Neural Computation. Addison-Wesley (1991)
3. Richards J.A.: Remote sensing digital image analysis: an introduction (1993)
4. Kohonen, T.: Self-Organizing Maps. Springer, Heidelberg, Germany (1997)
5. Specht D.F.: A general regression neural network. IEEE Transactions on Neural Networks 2(6)(1991) 568-576
6. Hyun B.G., Nam K.: Faults diagnoses of rotaing machines by using neural nets: GRNN and BPNN. Proceedings of the 1995 IEEE IECON 21st International Conference on Industrial ECI. Orlando (1995)
7. Cigizoglu H.K., Alp M.: Generalized regression neural network in modelling river sediment yield. Advances in Engineering Software 37(2006) 63-68

Comparison of One-Class SVM and Two-Class SVM for Fold Recognition

Alexander Senf, Xue-wen Chen, and Anne Zhang

The University of Kansas, Lawrence KS 66045 USA
ajsenf@ku.edu, xwchen@ku.edu, yazhang@eecs.ku.edu

Abstract. The best protein structure prediction results today are achieved by incorporating initial structural prediction using alignments to known protein structures. The performance of these algorithms directly depends on the quality and significance of the alignment results. Support Vector Machines (SVMs) have shown great potential in providing good alignment results in cases where very low similarities to known proteins exist. In this paper we propose the use of a one-class SVM to reduce the computational resources required to perform SVM learning and classification. Experimental results show its efficiency compared to two-class SVM algorithms while producing results of similar accuracy.

1 Introduction

Functional protein annotation is generally dependent on knowledge of the three-dimensional structure of a protein. However, our knowledge about protein structures grows at a much slower rate than the discovery of new protein sequences. Protein sequences can be recovered with relative ease from DNA sequences, but to determine the accurate structure of a protein with a given sequence still requires time-consuming experiments, such as X-Ray Crystallography or NMR. Determining the structure of a protein from its sequence computationally is among the most important problems in bioinformatics today.

Many methods have been developed over the past 25 years to generate structural information for unknown proteins by comparing their sequences to the sequences of proteins with known structures. A high degree of sequence similarity has been shown to entail a high degree of structural similarity, which also entails functional similarities. Databases such as the Structural Classification of Proteins (SCOP) database [11] have been devised to organize proteins according to various levels of structural and functional similarities.

One of the methods used to detect protein structures from sequences is called fold recognition. With fold recognition the unknown protein sequence is aligned to known proteins, and the statistical significances of the alignments are estimated. The sequence and location of secondary structural elements can then be determined using the information from the most significant matches in the database. This method is often combined with comparative modeling approaches to build a complete three-dimensional protein structure from the fragments predicted using fold recognition.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 140 – 149, 2006.

While fold recognition works well for proteins with high degrees of sequence similarity to known proteins, this method fails to produce satisfactory results if the closest known proteins exhibit less than 20% similarity. In these cases the best results are currently produced using molecular dynamics approaches, which physically simulate the folding process of the unknown protein. These methods, however, are computationally very intensive and are not yet able to produce sufficiently accurate results that could be used for functional annotation.

A better solution is to find known proteins that may be structurally and functionally related to the unknown sequence, even in the absence of any significant sequence similarities. The existence of such distant relationships has become apparent as the number of proteins with known structure increased in recent years, and an increasing number of proteins with similar structure and function were observed to have very low primary sequence similarities.

Among the methods developed to detect more distant similarity relationships are the use of sequence profiles as in PSI_BLAST [1], or profile Hidden Markov Models (HMM) [7]. Profiles extend the sequence similarity search from individual sequences to sequences families by incorporating statistical information from multiple sequence alignments of highly related proteins. Some methods have been developed to include structural information along with sequential information to increase the probability of finding remote structural relationships. Among these approaches are GenTHREADER [6] and 3D-PSSM [8].

A more recent promising attempt has been to combine the alignment of sequence profiles with support vector machine (SVM) classifiers [5,9,14]. Approaches such as in [14], which focus on the kernel function used with the SVM, or in [9], which combine pairwise sequence alignments with SVM classification indicate a significant improvement over conventional methods in the ability to detect remotely related proteins. The work of Han et al in [5] extends this approach by combining profile alignments with SVM classifiers.

In this paper we introduce a one-class SVM for the fold recognition problem. One-class SVMs offer significant savings in terms of space and speed over two-class SVMs, because only positive examples are needed to train the SVM. Previous applications of one-class SVMs in the area of Bioinformatics include [16] and [20]. One-class SVMs find their primary use in the field of novelty detection [17] or in document and image retrieval systems [4], [12], and have been shown to be capable of producing similar results as two-class SVMs [18].

The next section of this paper presents an overview of the theory behind two-class SVMs, and the adaptations for one-class SVMs. Section 3 describes the experimental setup and data used. Section 4 reports the results obtained performing the algorithm presented. The last section summarizes what can be learned from this experiment and gives a brief outlook on future research directions.

2 Support Vector Machines

2.1 Theory

An SVM is a machine learning technique based on Vapnik's Statistical Learning Theory [21]. Two-class support vector machines learn to distinguish between two classes in a given data set by fitting a hyperplane that maximally divides both classes.

This works well for data sets that are linearly separable. In cases where the data is not linearly separable, a linear SVM may still be used when it allows for a certain amount of errors. This is achieved by introducing a slack variable ξ and an upper bound C for the number of errors. The formula to be minimized then takes on the commonly used form:

$$J(\overline{w}, b, \overline{\xi}) = \tfrac{1}{2}(\overline{w} \cdot \overline{w}) + C\sum_{i=1}^{n} \xi_i \tag{1}$$

subject to $y_i\left[\overline{w} \cdot \overline{x_i} + b\right] \geq 1 - \xi_i$ and $\xi_i \geq 0$.

If the data points are not easily separable even with the provision for a certain amount of errors, the data can be projected into a higher-dimensional feature space using kernels. A kernel is a function that takes the original data points and several parameters, and increases their dimensionality. A good choice of kernel function and corresponding parameters will allow the data to then be separable by a hyperplane, using the same function described in (1). Examples of popular kernel functions are:

Polynomial,

$$\kappa(\overline{x}, \overline{y}) = (\overline{x} \cdot \overline{y})^d \tag{2}$$

Radial Basis Function (RBF),

$$\kappa\left(\overline{x}, \overline{y}\right) = \exp\left(\frac{-\left|\overline{x} - \overline{y}\right|^2}{(2\sigma)^2}\right) \tag{3}$$

and Sigmoid

$$\kappa\left(\overline{x}, \overline{y}\right) = \tanh\left(\kappa\left(\overline{x} \cdot \overline{y}\right) + \Theta\right) \tag{4}$$

with gain κ and offset Θ.

2.2 One-Class SVM

One-class SVMs were first proposed by Schölkopf in [15]. One-class SVMs are an extension of the original two-class SVM learning algorithm to enable the training of a classifier in the absence of any negative example data. Training can be achieved by treating a certain number of data points of the positive class as if they belong to the negative class. The idea is to define a boundary between the majority of the positive data points and outliers (or atypical data points). One-class SVMs use the parameter ν (Nu) to define the trade-off between the percentage of data points treated as the positive class and the negative class. Two approaches to generate this separating boundary are typically available:

The first approach to train a one-class SVM is to describe a classification function that conforms to a hypersphere boundary between the positive class and the outliers, based on a density distribution function. The parameter ν determines the shape of the boundary.

The second approach fits a hyperplane between the origin (of the coordinate system) and the data points, separating a certain percentage of outliers from the rest of the data points. This approach has been shown to be equivalent to the decision hypersphere and is used by many one-class SVM implementations due to its simpler implementation. The LIBSVM package uses this approach.

These requirements for the separation boundary can be formulated mathematically by providing a measure *f*(*z*) of the distance *d*(*z*) to the positive class, or of the probability *p*(*z*) of belonging to the positive class, and a threshold *θ* to distinguish between the positive class and the outliers [18]:

$$f(z) = I(d(z) < \Theta_d) \tag{5}$$

or

$$f(z) = I(p(z) > \Theta_d) \tag{6}$$

where *I* is the function indicating positive or negative class membership. One-class classifiers learn by optimizing the function *d*(*z*) or *p*(*z*). Some implementations go on to optimize the parameter *θ* while some use an a-priori defined threshold, usually provided by the parameter ν.

The error rate of a one-class SVM is estimated as the fraction of the positive (target) class f_{T+} versus the fraction of outliers that is rejected f_{O-}. The density distribution of f_{O-} needs to be estimated to calculate this error measure. Error E_I then is the group of positive examples rejected by the classifier, and E_{II} is the group of negative examples accepted by the classifier.

Various methods are available to estimate a distribution for *Z*: density methods, boundary methods, and reconstruction methods. The most simple of these is a Gaussian density model with a probability distribution as:

$$p_N(z;\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}\left|\Sigma^{1/2}\right|}\exp\left\{-\tfrac{1}{2}(z-\mu)^T\Sigma^{-1}(z-\mu)\right\} \tag{7}$$

where *μ* is the mean, Σ the covariance matrix, and *d* the dimension of an object. More complicated probability and density functions have been studied as well. A notable difference with boundary methods is that these methods allow for multiple boundaries to cover a given positive class, but the parameters for the number of distinct boundaries need to be supplied.

Kernel functions can be applied to one-class SVM data points in the same way as for two-class SVMs, allowing more complicated data sets to be used with one-class SVMs.

2.2 Discussion: SVM and One-class SVM

The primary difference between a two-class SVM and a one-class SVM is the use of negative data points in the training of a classification functions. The one-class SVM approach has the advantage of being able to use a very small training set to learn a classification function.

This feature has been used successfully in [20] to study small-size genomes. When dealing with genes or proteins there already exists a very large amount of known data. It is not always desirable to use the entire dataset available to train classification functions for certain features or sequences. Additionally, it is also not easy to decide which of the data is relevant and which data can be left out when designing a classifier. The one-class SVM approach allows for a solution, as it only requires the data of the class to be discovered to learn a decision function. This allows for substantial savings in computation time and memory space, while maintaining a comparable level of accuracy.

3 Data Set and Feature Extraction

The algorithm presented in this paper attempts to determine structurally and functionally related protein domains at three different levels in the SCOP hierarchy: family, superfamily, and fold. The data for the experiments is taken from SCOP version 1.65. Using the subset of domains with <40% pairwise similarity from the ASTRAL compendium [2], all folds with at least 40 members are initially selected. This results in a data set with 1870 domains, comprised of 5 classes and 21 different folds. Each selected fold contains, on average, more than 10 superfamilies. This set is randomly divided into two-thirds training data and one-third testing data, resulting in 1247 training sequences and 623 testing sequences, representing members of all folds in each set.

3.1 SVM Training

This algorithm initially creates sequence profiles running six iterations of PSI-BLAST, generating the Position Specific Scoring Matrices (PSSM) and Position Frequency Matrices (PFM) for each training and testing sequence. Each profile is generated using PSI-BLAST default settings, except for the number of iterations, which is specified to be six (j=6). The PSSM and PFM output is stored to disk using parameter Q. Next, an all-against-all alignment of profiles in the training data is created. The alignment matrix m_{ij} of a profile q and a profile t is given by

$$m_{ij} = \sum_{k=1}^{20} \left[f_{ij}^{q} S_{jk}^{t} + S_{ij}^{q} f_{jk}^{t} \right] \tag{8}$$

where f, S are the PFM scores and PSSM scores of amino acid k at position i of profile q or at position j of profile t, respectively [5].

These 1246 m_{ij} alignment matrices for profile i (i=1, 2, …, 1247) of length n_i are used to extract (n+1) dimensional feature vectors s=(sa^1, sa^2, …, sa^n, total_score), where total_score is the total score of the alignment. The alignment feature vectors are then smoothed using

$$sa^i = m^{i-2} + 2m^{i-1} + 3m^i + 2m^{i+1} + m^{i+2} \tag{9}$$

where m^i is the profile alignment score at position i [19]. To produce comparable feature vectors, the individual scores are scaled down to values between 0 and 1.

Feature vectors of alignments between profile *i* and profiles of the same fold as *i* are assigned the class label 1 (positive class), all other feature vectors are assigned class label 0 (negative class).

While the one-class SVM used in this approach only requires feature vectors belonging to the positive class, the full data set is computed to enable a performance comparison with the two-class SVM. Two feature vector files are produced: one containing only the positive class for each profile, and one containing positive and negative examples to be used with the two-class SVM.

The SVMs are then trained using the radial basis function (RBF) kernel. The training algorithm performs some basic parameter optimization for each of the 1247 two-class SVMs to be trained, and produces a trained model for each profile. For the one-class SVM, the parameter ν is set to the default value of ν =0.5, which indicates that 50% of the positive class feature vectors are treated as outliers. An optimization algorithm is used to determine the RBF kernel parameter gamma.

The freely available software LIBSVM [3] is used for all SVM training and testing. LIBSVM is available for download at http://www.csie.ntu.edu.tw/~cjlin/libsvm/. All programs written for this algorithm are implemented in Java, incorporating some of the LIBSVM Java source code. A task distribution system written in Fortran is used to distribute individual tasks over multiple CPUs. The algorithm was executed on a parallel machine comprised of dual and quad Intel Xeon 3200 EMT64 nodes communicating with MPICH2 under Linux.

3.2 SVM Testing

PSSM and PFM matrices are generated for the testing data in the same way as for the training data. Aligning each test profile with all training profiles generates a set of 1247 feature vectors for every test profile. These feature vectors are then evaluated with the corresponding trained SVM models. A score is produced for each result by summation of all positive evaluations. The training sequences with the highest-scoring results are taken as candidate targets for the test sequence.

3.3 Algorithm Flowchart

Start with ASTRAL subset of domains with <40% similarity.

1. Select folds with at least 40 members.
2. Use PSI-BLAST to generate profiles matrices (PSSMs and PFMs).
3. Divide data set: 2/3 training data, 1/3 testing data.
4. Training Data: Perform all-to-all profile-profile alignments. For the one-class SVM, only align within each fold group. For the two-class SVM, align all training profiles.
5. Training Data: Extract feature vectors from alignments, and scale and smooth feature vectors.
6. Training Data: Train one-class SVM and two-class SVM for each set of feature vectors.
7. Testing Data: Perform one-to-one profile-profile alignment between each test profile and all training profiles, producing 1247 feature vectors for each testing sequence (same for one-class SVM and two-class SVM).

8. Testing Data: Extract feature vectors from alignments, and scale and smooth feature vectors.
9. Testing Data: Evaluate feature vectors of each testing profile with all appropriate SVM models.
10. Testing Data: Sum results of each evaluation. Rank groups of results for each testing profile to get highest-scoring results.

This flowchart outlines the essential flow of data through our algorithm. Steps 4, 5, 6, 7 and 8 can be performed in parallel. Step 9 requires step 6 to be completed, as it needs the output of step 6 as input.

3.4 Performance Assessment

The results obtained in this paper compare the performance of a two-class SVM with the performance of a one-class SVM approach. The algorithm presented builds upon the work of Han et al in [5]. Han et al. used the same data set from the ASTRAL compendium version 1.65, selecting all folds with at least 20 domains, resulting in 62 folds and 2,854 domains. This data was pre-processed using the same algorithm as this paper. Two-class SVMs were trained using an RBF kernel, and the results were post-processed to produce statistically comparable results between all SVM classifications.

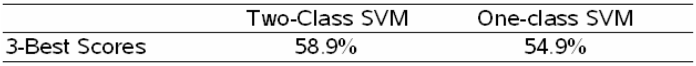
4 Experimental Results

Parameter selection is very important with SVM learning algorithms. This algorithm performs a simple grid-searching parameter optimization routine with 8-fold cross validation to select (LIBSVM-) SVM parameters *c* (upper bound in the number of errors allowed to occur) and RBF kernel parameter *gamma* for the two-class SVM learning process. The one-class SVM learning routine uses a basic algorithm to optimize the RBF kernel parameter *gamma*. The SVM parameter *g*, which determines the fraction of training data points to include in the positive class, is left at the LIBSVM default value of 0.5.

Running this algorithm with the two-class SVM, and counting only the top-three scoring results as candidate solution produces an accuracy rate of 58.9%, which means that 58.9% of tested sequences contain the correct fold among the top-three scoring results. Running the same algorithm using the one-class SVM instead produced an accuracy rate of 54.9%. It is notable that very little parameter optimization has yet been attempted for the one-class SVM.

One-class SVMs require less time and storage space to run compared to two-class SVMs. For this algorithm, computational savings for the one-class SVM occur when

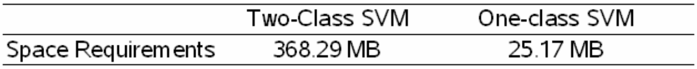
Table 1. Classification performance



	Two-Class SVM	One-class SVM
3-Best Scores	58.9%	54.9%

generating the feature vectors, during parameter optimization, and SVM training. Significant space and time savings of the one-class SVM algorithm over two-class SVMs can be expected in steps 4, 5, and 6 of the algorithm described above. Comparing the space required of the algorithm, the two-class SVM requires 368.29MB to store the training feature vector files, compared to 25.17MB for the one-class SVM algorithm. This is a savings of 1 order of magnitude, and is due to the fact that the one-class SVM does not require any negative examples to be generated and stored. The evaluation of the test sequences is identical between the one-class SVM and two-class SVM approaches.

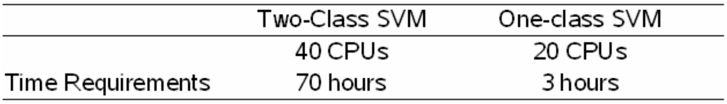
Table 2. Space requirements for training feature vector files



	Two-Class SVM	One-class SVM
Space Requirem ents	368.29 MB	25.17 MB

The advantages regarding time are due partly to the sufficiency of positive examples, which speeds up the feature vector generation phase, and partly because the training step (step 6) for the one-class SVM is completed faster than for the two-class SVM. This second component of savings varies with the kernel function used, and is much more pronounced using a linear kernel as compared with the RBF kernel. Utilizing 20 CPUs, the one-class SVM training step was completed in 3 hours. By comparison, the two-class SVM training step, utilizing 40 CPUs, took 70 hours.

Table 3. CPU time requirements for SVM learning



	Two-Class SVM	One-class SVM
	40 CPUs	20 CPUs
Time Requirements	70 hours	3 hours

These improvements regarding space and time requirements enable the one-class SVM based algorithm to scale much easier to larger data sets. Adding more sequences to the training data only affects the subset of SVMs for the same fold as the new sequences. The two-class algorithm requires feature vector generation, and re-training, for all training SVMs.

5 Conclusions and Future Work

In this paper we introduced a one-class SVM approach to detect remote relationships between protein sequences with very low sequence similarities. The algorithm begins by generating sequence profiles for all training sequences using PSI-BLAST. In the next step profile-profile alignments between each training sequence and all other training sequences are generated. For the one-class SVM algorithm, only alignments between sequences belonging to the same fold are required. The two-class SVM algorithm assigns all sequences belonging to the same fold to the positive class, and all remaining sequences to the negative class. This results in a set of feature vectors for each training sequence. Individual SVMs are then trained for each training sequence, using either a two-class SVM or a one-class SVM and using the Radial Basis Function kernel.

The advantages of the one-class SVM approach become apparent when comparing the reduced time and space requirements, which show a significant improvement over the two-class SVM approach.

The algorithm presented in this paper produces initial protein sequence profiles by running six iterations of PSI-BLAST, and using the resulting PSSM and PFM matrices to perform profile-profile alignments. A different approach to profile-profile alignments has been given in the recent literature by Söding in [16], where alignments are produced using profile HMMs instead of PSSMs and PFMs. This approach may well be capable of improving the performance of this algorithm.

A second focus of improvement lies in the kernel function used for SVM learning and classification. Several papers in the recent literature have presented novel kernel functions aimed at improving the ability to detect remote sequence relationships, as for example in [13]. This offers a second promising direction of future research to improve the performance of this algorithm.

References

1. Altschul, S. F, et al.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research Vol. 25. (1997) 3389-3402.
2. Brenner SE, Koehl P, Levitt M.: The ASTRAL compendium for sequence and structure analysis. Nucleic Acids Research. Vol. 28. (2000) 254-256.
3. Chang, Chih-Chung and Chih-Jen Lin: LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/. (2001).
4. Chen, Yunquiang, Zhou, Xiang, and Thomas S. Huang. One-Class SVM For Learning in Image Retrieval. Proc. IEEE Int'l Conf. On Image Processing (ICIP). 7 Oct.-10 Oct. 2001. Vol. 1. (2001) 34-37.
5. Han, S., B.-c. Lee, et al.: Fold recognition by combining profile-profile alignment and support vector machine. Bioinformatics. Vol. 21(11). (2005) 2667-2673.
6. Jones, D.T.: GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. Journal of Molecular Biology. Vol. 287. (1999) 797-815.
7. Karplus, K., et al.: Predicting protein structure using only sequence information. Proteins. Suppl 3. (1999) 121-125.
8. Kelly, L. A., et al.: Enhanced genome annotation using structural profiles in the program 3D-PSSM. Journal of Molecular Biology. Vol. 299. (2000) 499-520.
9. Liao, Li, and William Stafford Noble.: Combining Pairwise Sequence Similarity and Support Vector Machines for Detecting Remote Protein Evolutionary and Structural Relationships. Journal of Computational Biology. Vol. 10(6). (2003) 857-868.
10. Manewitz, Larry M., and Malik Yousef.: One-Class SVMs for Document Classification. Journal of Machine Learning Research. Vol. 2(3). (2001) 139-154
11. Murzin A. G., Brenner S. E., Hubbard T., Chothia C.: SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. Vol. 247. (1995) 536-540.

12. Onoda, Takashi, Murata, Hiroshi, and Yamada, Seiji. One Class Support Vector Machine based Non-Relevance Feedback Document Retrieval. Proc. IEEE Int'l Joint Conf. on Neural Networks (ICJNN). 31 July-4 Aug. 2005. Vol. 1. (2005) 552-557.
13. Rangwala, Huzefa, and George Karpys. Profile-based direct kernels for remote homology detection and fold recognition. Bioinformatics. Vol. 21(23). (2005) 4239-4247.
14. Saigo, Hiroto, et al.: Protein homology detection using string alignment kernels. Bioinformatics. Vol. 20(11). (2004) 1682-1689.
15. Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., Williamson, A.: Estimating the support for a high-dimensional distribution. Microsoft Research, One Microsoft Way Redmond WA 98052, Tech. Rep. MSR-TR-99-87, 1999.
16. Söding, Johannes. Protein homology detection by HMM-HMM comparison. Bioinformatics. Vol. 21(7). (2005) 951-960.
17. Spinosa, Eduardo J. and de Carvalho, Andre C.P.L.F.: Support vector machines for novel class detection in Bioinformatics. Genet. Mol. Res. Vol. 4(3). (2005) 608-615.
18. Tax, David M. J.: One-class classification-concept learning in the absence of counter-examples. Ph.D. Dissertation, Delft University of Technology. ASCI Dissertation Series. Vol. 65. (2001) 1-190
19. Tress, M. L., et al.: Predicting reliable regions in protein alignments from sequence profiles. Journal of Molecular Biology. Vol. 330. (2003) 705-718.
20. Tsirigos, Aristotelis and Isidore Rigoutsos: A sensitive, support-vector-machine method for the detection of horizontal gene transfers in viral, archaeal and bacterial genomes. Nucleic Acids Research. Vol. 33(12). (2005) 3699-3707.
21. Vapnik, V. N. (1995) The Nature of Statistical Learning Theory. Springer-Verlag, New York.
22. Zhang, Jianguo, Ma, Kai-Kuang, Er, Meng Hwa, and Vincent Chong. Tumor Segmentation from Magnetic Resonance Imaging by Learning Via One-Class Support Vector Machine. International Workshop on Advanced Image Technology (IWAIT04). (2004) 207-211.

Efficient Domain Action Classification Using Neural Networks

Hyunjung Lee[1], Harksoo Kim[2], and Jungyun Seo[3]

[1] Natural Language Processing Lab., Department of Computer Science, Sogang University,
1 Sinsu-dong, Mapo-gu, Seoul, 121-742, Republic of Korea
juvenile@sogang.ac.kr
[2] Program of Computer and Communications Engineering, College of Information Technology,
Kangwon National University, 192-1, Hyoja 2(i)-dong, Chuncheon-si, Kangwon-do, 200-701,
Republic of Korea
nlpdrkim@kangwon.ac.kr
[3] Department of Computer Science and Interdisciplinary Program of Integrated Biotechnology,
Sogang University, 1 Sinsu-dong, Mapo-gu, Seoul, 121-742, Republic of Korea
seojy@sogang.ac.kr

Abstract. Speaker's intentions can be represented into domain actions (domain-independent speech acts and domain-dependent concept sequences). Therefore, domain action classification is very useful to a dialogue system that should catch user's intention in order to generate correct reaction. In this paper, we propose a neural network model to determine speech acts and concept sequences at the same time. To avoid biased learning problems, the proposed model uses low-level linguistic features and filters out uninformative features using χ^2 statistic. In the experiment, the proposed model showed better performances than the previous work in speech act classification. Moreover, the proposed model showed meaningful results when the size of training corpus was small. Based on the experimental results, we believe that the proposed model will be more helpful to dialogue systems because it manages speech act classification and concept sequence classification at the same time. We also believe that the proposed model can alleviate sparse data problems in speech act classification.

1 Introduction

A goal-oriented dialogue consists of a sequence of goal-oriented utterances. Speakers' intentions indicated by goal-oriented utterances can be represented by shallow semantic forms called domain actions [1], [7]. As shown in Table 1, a domain action consists of a pair of a speech act and a concept sequence. The speech act represents the general intention expressed in an utterance, and the concept sequence captures the semantic focus of the utterance.

If we plan to implement an intelligent dialogue system, we should first prepare a domain action identification module because users' intentions can be captured by domain actions. However, it is difficult to directly infer domain actions from surface utterances because the domain actions depend on the contexts of the utterances. For

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 150 – 158, 2006.

Table 1. An example of utterances along with their corresponding domain actions; *S* means a system, and *U* means a user

Utterance	Domain action
(1) User: Hello.	Greeting & NULL
(2) S: May I help you?	Opening & NULL
(3) U: Tell me the tomorrow schedule.	Request & Timetable-search
(4) S: You have an appointment with Kildong Hong at 11 a.m.	Response & Timetable-search
(5) U: We changed the appointment.	Inform & Timetable-modify
(6) S: What is changed?	Ask-ref & Timetable-modify
(7) U: The appointment date was changed.	Response & Timetable-modify-date
(8) S: When is the changed date?	Ask-ref & Timetable-modify-date
(9) U: It's December 5.	Response & Timetable-modify-date

example, the domain action of utterance (9) in Table 1 can be 'inform & timetable-search-date' and 'response & timetable-modify-date' in surface analysis. To resolve this ambiguity, the dialogue system should analyze the context of utterance (9). In this case, checking the previous utterance, i.e., utterance (8), is necessary for choosing 'response & timetable-modify-date' as the domain action of utterance (9).

Previous approaches for identification of users' intentions have been based on knowledge such as recipes for plan inference and domain specific knowledge [2], [4], [8]. These models depend on costly handcrafted knowledge so that it is difficult to scale up and expand them to other domains. To overcome this problem in recent years, there has been an increased interest in using machine learning models for the processing of users' utterances [5], [6], [10]. The machine learning models offer a means of associating features of utterances with particular categories indicating users' intentions, since the computer can efficiently analyze a large quantity of data and consider many different feature interactions. However, the machine learning models are critically affected by the performances of underlying feature selection systems. If the input features are slightly biased by analysis errors of underlying feature selection systems, they do not take a full advantage of particular features of utterances which may provide valuable clues for identifying users' intentions on account of biased learning.

In principle, neural networks can compute any computable function, i.e., they can do everything a normal digital computer can do. Especially anything that can be represented as a mapping between vector spaces can be approximated to arbitrary precision by feed-forward neural networks (which are the most frequently used type). In practice, neural networks are especially useful for solving mapping problems to which hard and fast rules cannot be easily applied. In spite of the advantage, we could not easily find nice application systems using neural networks on natural language processing. We think that the reason is the absence of effective feature selection methods. Effective feature selection is significant since it increases tagging performance and decreases training time. Therefore, automatic and effective feature selection methods are requested.

In this paper, we propose a domain action classification model using neural networks in a schedule management domain. To reduce biased learning errors, the model does not use syntactic and semantic features as input features, but the model uses only low-level linguistic features [12] such as lexicals and POS's (parts-of-speech) because morphological analyzers generally make much less errors than syntactic parsers and semantic analyzers. In addition, to automatically select informative linguistic features, the model adopts χ^2 statistic because the feature selection method showed better results than mutual information and information gain in text categorization [14]. The current version of the proposed model operates in Korean, but we believe that language conversion will not be a difficult task because the model uses shallow natural language processing techniques.

This paper is organized as follows. In Section 2, we propose a domain action classification model in which a speech act classification model and a concept sequence classification model are integrated. In Section 3, we explain experimental setup and report some experimental results. Finally, we draw some conclusions in Section 4.

2 Domain Action Classification Using Neural Networks

We design two neural network models for domain action classification. One is a concept sequence classification model, and the other is a speech act classification model. We call the concept sequence classification model CSCM and the speech act classification model SACM. We call CSCM and SACM together the domain action classification model (DACM). Fig. 1 shows the architectures of DACM.

As shown in Fig. 1, the inputs of the proposed models are divided into two parts; sentential feature part and contextual feature part. The sentential feature part represents the relationships between the speech acts (or concept sequences) and the surface

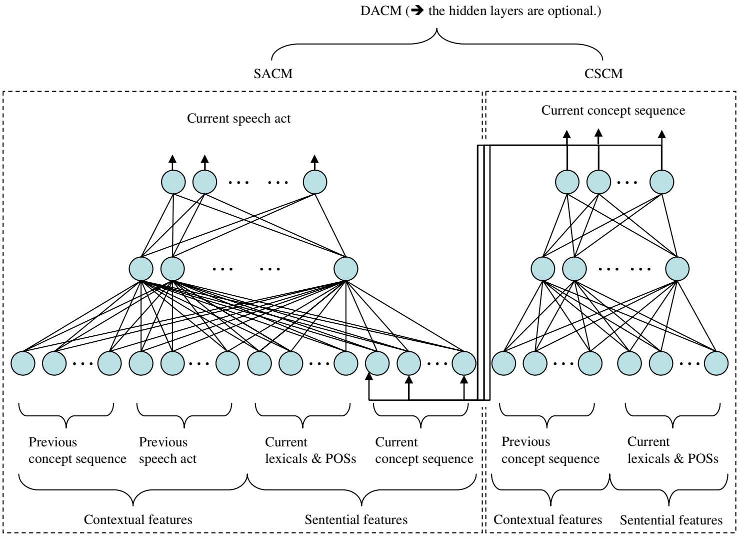


Fig. 1. The architecture of DACM

sentences. However, it is impossible to use surface sentences as input features of neural networks because a speaker expresses identical contents with various surface forms of sentences according to a personal linguistic sense in a real dialogue. To overcome this problem, we assume that an utterance can be generalized by a set of sentential features. The sentential feature part of CSCM consists of two components; lexical features (content words annotated with POS's) and POS features (POS bi-grams of all words in an utterance). On the other hand, the sentential feature part of SACM consists of three components; lexical features, POS features, and concept sequence features. Generally, content words include nouns, verbs, adjectives and adverbs, while functional words involve prepositions, conjunctions and interjections. For example, in SACM, the sentential feature set of utterance (9) in Table 1 consists of two lexical features, four POS features, and a concept sequence feature, as shown in Fig. 2.

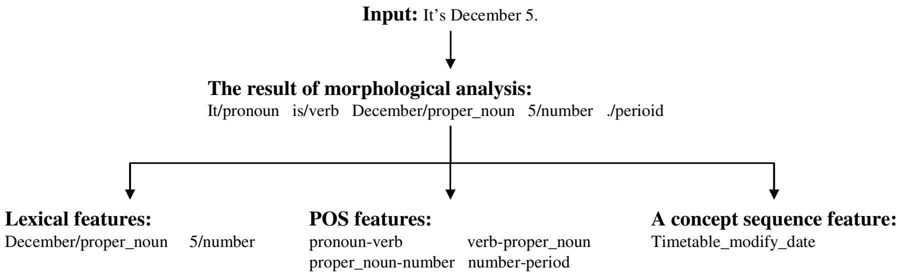

Input: It's December 5.
The result of morphological analysis:
It/pronoun is/verb December/proper_noun 5/number ./perioid
Lexical features:
December/proper_noun 5/number
POS features:
pronoun-verb verb-proper_noun
proper_noun-number number-period
A concept sequence feature:
Timetable_modify_date

Fig. 2. An example of a sentential feature set in SACM

The reason why we use an additional feature, the concept sequence feature, for speech act classification is as follows. Although the lexical features and the POS features may offer informative clues (e.g. the interrogatives like *what* and *who* can be very important clue words for speech act classification) to SACM, we think that these features cannot fully contain the contents[1] of utterances. We found that we should sometimes consider the concept sequences of utterances in order to determine correct speech acts, as shown in Table 2.

Table 2. An example why we should consider concept sequences for speech act classification; *S* means a system, and *U* means a user

Utterance	Concept sequence
(1) S: What was changed?	Timetable-modify
(2) U: The appointment date was changed.	Timetable-modify-date
(3) U: It's December 5.	Timetable-modify-date

In Table 1, utterance (3) has several surface speech acts such as *inform* and *response*. Such an ambiguity can be solved by considering the concept sequence of

[1] In this paper, we approximate the contents of utterances to their concept sequences.

utterance (3). If we consider only the speech act of utterance (2) to determine the speech act of utterance (3), the speech act of utterance (3) may be *inform*. However, if we consider the concept sequence of utterance (3), we can find that utterance (3) is closely associated with utterance (2) and its meaning is the changed date. Based on these facts, we can find that the speech act of utterance (3) is *response* for the user to give the system additional information (*cf.* "The appointment date was changed to December 5"). To obtain the concept sequence of current utterance, SACM uses the output of CSCM.

To obtain the lexical features and POS features, we use a conventional morphological analyzer. Then, we remove non-informative features by using a well-known χ^2 statistic because the previous works in document classification have shown that effective feature selection can increase precisions [9], [11], [13], [14]. The χ^2 statistic measures the lack of independence between a feature f and a category c (in this paper, a speech act or a concept sequence), as shown in Equation (1).

$$\chi^2(f,c)=\frac{(A+B+C+D)\times(AD-CB)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)} \tag{1}$$

In Equation (1), A is the number of times f and c co-occur, B is the number of times f occurs without c, C is the number of times c occurs without f, and D is the number of times neither c nor f occurs. To remove non-informative features, we calculate the feature scores as the maximum χ^2 statistic of a feature-category pair, as shown in Equation (2), and choose top-n features according to the feature scores.

$$\chi^2_{\max}(f)=\max_{i=1}^{m}\{\chi^2(f,c_i)\} \tag{2}$$

The contextual feature part represents the relationships between a current speech act (or a current concept sequence) and previous speech acts (or previous concept sequences). Since it is impossible to consider all previous speech acts (or all previous concept sequences) as contextual information, we use the bi-gram model, as shown in Fig. 1. By the same reason with the sentential feature part of SACM, the contextual feature part of SACM consists of a previous speech act and a previous concept sequence. On the other hand, the contextual feature part of CSCM consists of only a previous concept sequence.

3 Evaluation

3.1 Data Sets and Experimental Settings

We collected a Korean dialogue corpus simulated in a schedule management domain such as appointment scheduling and alarm setting. The dialogue corpus were obtained by eliminating interjections and erroneous expressions from the original transcriptions of simulated dialogues between two speakers to whom a task of the dialogue had been given in advance: one participant freely asks something about his/her daily schedules, and the other participant responds to the questions or asks back some questions by using knowledge bases given in advance. This corpus consists of 956 dialogues,

Table 3. A part of the annotated dialogue corpus

Tag	Values	Tag	Values
/ID/	4-5	/ID/	4-7
/SP/	User	/SP/	User
/KS/	약속 시간이 몇 시지?	/KS/	장소는 어디야?
/EN/	When is the appointment time?	/EN/	Where is the place?
/SA/	Ask-ref	/SA/	Ask-ref
/CS/	Timetable-search-time	/CS/	Timetable-search-place
/ID/	4-6	/ID/	4-8
/SP/	System	/SP/	System
/KS/	11시 30분입니다.	/KS/	코엑스홀입니다.
/EN/	It’s eleven thirty.	/EN/	It’s COEX Hall.
/SA/	Response	/SA/	Response
/CS/	Timetable-search-time	/CS/	Timetable-search-place

21,336 utterances (22.3 utterances per dialogue). Each utterance in dialogues was manually annotated with speech acts and concept sequences. Table 3 shows a part of the annotated dialogue corpus.

In Table 3, KS represents a Korean sentence and EN represents the translated English sentence that is not unseen in the original dialogue corpus. SP has a value of either User or System depending on the speaker. The manual tagging of speech acts and concept sequences was done by five graduate students with the knowledge of a dialogue analysis and post-processed by a student in a doctoral course for consistency.

In order to experiment the proposed model, we divided the annotated dialogue corpus into the test corpus with 100 dialogues and the training corpus with 856 dialogues. We again divided the training corpus into 8 parts (100, 200, …, 700, 856 dialogues) to compare the precisions as the size grows up. The types of speech acts are very subjective without an agreed criterion, and the types of concept sequences depend on application domains. In this paper, we defined 11 types of speech acts and 53 types of concept sequences. Table 4 shows the speech acts that we defined.

Table 4. Speech acts and their meanings

Speech act	Description	Example
Greeting	The opening greeting of a dialogue	Hello.
Expressive	The closing greeting of a dialogue	Good-bye.
Opening	Sentences for opening a goal-oriented dialogue	May I help you?
Ask-ref	WH-questions	Where is the place?
Ask-if	YN-questions	Can I change the time?
Response	Responses of questions or requesting actions	Yes, you can.
Request	Declarative sentences for requesting actions	Set the alarm.
Ask-confirm	Questions for confirming the previous actions	Saturday, right?
Confirm	Reponses of ask-confirm	Right.
Inform	Declarative sentences for giving some information	It was canceled.
Accept	agreement	I know.

In the experiments, we set the number of sentential features except concept sequence features to 100 in total. In other words, we selected top-100 features using χ^2 statistic. The learning rate of the proposed model was 0.2, and trainings spent 200 epochs.

3.2 Experimental Results

To evaluate the performances of the proposed model according to the various sizes of training corpus, we calculated the precisions of the proposed model at various cutoff points, as shown in Table 5 and Fig. 3.

Table 5. The precisions of domain action classification

The size of training corpus	Speech act classification		CSCM
	SACM	Kim-2004	
100	81.09	79.97	66.25
200	78.36	77.96	68.84
300	83.10	79.88	72.33
400	82.34	80.06	71.75
500	84.04	81.85	72.82
600	82.83	81.22	71.93
700	84.00	81.67	73.58
856	86.05	82.79	73.76

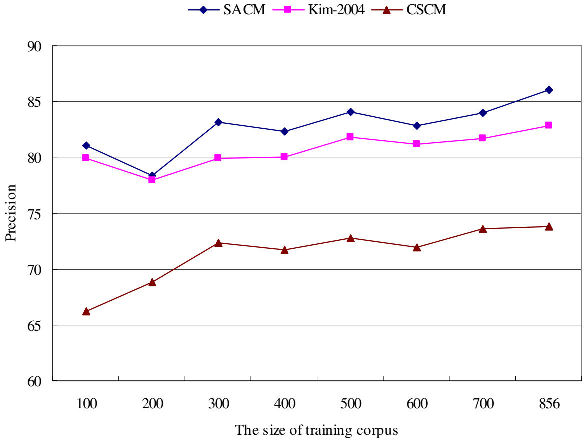


Fig. 3. The precisions of domain action classification in graph

In Table 5, Kim-2004 [3] is similar to SACM except that Kim-2004 does not use the concept sequence features as input features. As shown in Table 5, SACM showed better results than Kim-2004 at all cutoff points. Moreover, SACM using 100 dialogues as training corpus had similar precisions to Kim-2004 using 500~700 dialogues as training corpus. This fact shows that the concept sequence features are effective in determining speech acts. It also shows that SACM can alleviate sparse data problems in speech act classification. The precisions of CSCM were lower than those of SACM. We think that it was caused by the difference between the numbers of target categories: the target categories of SACM are 11 types of speech acts, but the target categories of CSCM are 53 types of concept sequences. Although CSCM does not perform well, DACM can be used as an essential module for a dialogue system because it outputs concept sequences as well as speech acts at the same time.

We analyzed the cases that DACM failed to return correct results. The failure reasons are as follows. First, DACM used a linearly adjacent speech act (or a linearly adjacent concept sequence) as contextual information. However, dialogues have hierarchical discourse structures, as shown in Fig. 4.

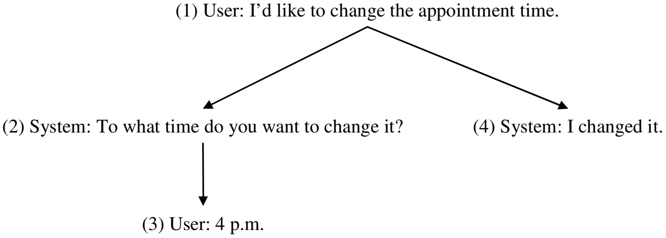

(1) User: I'd like to change the appointment time.
(2) System: To what time do you want to change it?
(4) System: I changed it.
(3) User: 4 p.m.

Fig. 4. An example of a hierarchical discourse structure

For example, if we want to know the domain action of utterance (4) in Fig. 4, we should consider not utterance (3) but utterance (1) as contextual information because utterance (4) is adjacent to utterance (1) in the tree structure of the discourse. To overcome this problem, we should study on methods to apply discourse structures to DACM. Second, the precisions of CSCM were much lower than those of SACM. The low precisions of CSCM affected the performances of SACM. When we used correct concept sequences as input features of SACM, the highest precision of SACM was 92%. Therefore, if we can improve the precisions of CSCM, SACM will perform much better.

4 Conclusion

We proposed a neural network model which can perform both speech act classification and concept sequence classification in Korean. To reduce biased learning errors, the proposed model uses low-level linguistic features such as lexicals and POS's as input features, and filters out uninformative input features using χ^2 statistic. After selecting features, the proposed model determined both speech acts and concept sequences at the same time using the same framework. In the experiment, the proposed model outperformed the previous work in speech act classification. Moreover, the proposed model showed meaningful results when we used small sizes of training corpus. Based on these experiments, we believe that the proposed model will be more helpful to dialogue systems than previous works (speech act classification models) because it manages speech act classification and concept sequence classification at the same time. We also believe that the proposed model can alleviate sparse data problems by using concept sequences as input features in speech act classification. In addition, we found that that neural networks can perform well on natural language processing insofar as effective methods is available for reducing the number of input features.

Acknowledgement

This research was supported as a Brain neuroinformatics Research program by Korean Ministry of Commerce, Industry, and Energy. It was also partially supported by Kangwon Institute of Telecommunications and Information (KITI).

References

1. Allen, J.: Natural Language Understanding. The Benjamin/Cummings Publishing Company, Inc., (1987)
2. Caberry, S.: A Pragmatics-based Approach to Ellipsis Resolution. Computational Linguistics, Vol. 15(2), (1989) 75-96
3. Kim, K., Kim, H., Seo, J.: A Neural Network Model with Feature Selection for Korean Speech Act Classification. International Journal of Neural Systems, Vol. 14 (6), (2004) 407-414
4. Lambert, L., Caberry, S.: A Tripartite Plan-based Model of Dialogue. In: Proceedings of ACL, (1991) 47-54
5. Langley, C.: Analysis for Speech Translation Using Grammar-based Parsing and Automatic Classification. In: Proceedings of the ACL Student Research Workshop, (2002)
6. Lee, S., Seo, J.: An Analysis of Korean Speech Act Using Hidden Markov Model with Decision Trees. In: Proceedings of ICPOL, (2001) 397-400
7. Levin, L., Langley, C., Lavie, A., Gates, D., Wallace, D., Peterson, K.: Domain Specific Speech Acts for Spoken Language Translation. In: Proceedings of 4th SIGdial Workshop on Discourse and Dialogue, (2003)
8. Litman, D. J., Allen, J. F.: A Plan Recognition Model for Subdialogues in Conversations. Cognitive Science, Vol. 11, (1987) 163-200
9. Lweis, D. D., Ringuette, M.: Comparison of Two Learning Algorithms for Text Categorization. In: Proceedings of SDAIR, (1994)
10. Samuel, K., Caberry, S., Vijay-Shanker, K.: Computing Dialogue Acts from Features with Transform-based Learning. In: Proceedings of the AAAI Spring Symposium, (1998) 90-97
11. Schűtze, H., Hull, D. A., Pedersen, J. O.: A Comparison of Classifiers and Document Representations for the Routing Problem. In: Proceedings of SIGIR, (1995)
12. Stolcke, A., Ries, K., Coccaro, N., Shiriberg, E., Bates, R., Jurafsky, D., Taylor, P., Van Ess-Dykema, C., Martin, R., Meteer, M.: Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. Computational Linguistics, Vol. 26(3), (2000) 339-373
13. Wiener, E., Pedersen, J. O., Weigend, A. S.: A Neural Network Approach to Topic Spotting. In: Proceedings of SDAIR, (1995)
14. Yang, Y., Pedersen, J. O.: A Comparative Study on Feature Selection in Text Categorization. In: Proceedings of the 14th International Conference on Machine Learning, (1997)

A New Hierarchical Decision Structure Using Wavelet Packet and SVM for Brazilian Phonemes Recognition

Adriano de A. Bresolin[1], Adrião Duarte D. Neto[2], and Pablo Javier Alsina[2]

[1] UTFPR - Technological Federal University of the Paraná – Brazil
Postal Box: 271, Av. Brasil, 4232
CEP 85.884-000, Medianeira, PR, Brazil
[2] UFRN - Federal University of the Rio Grande do Norte – Brazil
Postal Box: 1524, Campus Universitário Lagoa Nova
CEP 59072-970, Natal, RN, Brazil
{aabresolin, adriao, pablo}@dca.ufrn.br

Abstract. In this work, a new phonemes recognition system is proposed. The base of decision of the proposed system is the tongue position and roundedness of the lips. The features of the speech are the coefficients of Wavelet Packet Transform with sub-bands selected through the Mel scale. The SVM (Support Vector Machine) is used as classifier in the structure of a Hierarchical Committee Machine. The database used for the recognition was a set of oral vocalic phonemes of the Portuguese language. The experimental results show success rates of 97.50% for the user-dependent case and 91.01% for the user-independent case. This new proposal increased 3.5% the success rate in relation to the "one vs. all" decision strategy.

Keywords: Speech Recognition, Support Vector Machine, Wavelet Packet.

1 Introduction

A first decision in the development of a speech recognition system is the definition of the unit to be recognized: *words, syllables, triphones, diphones or phonemes.*

A natural language, such as the Portuguese, possesses about 400.000 words, what demands great amount of processing and storage, a hard problem for continuous recognition. In the last years, research efforts have focused the unit smaller than the word. Santos and Alcaim [10], used syllables as units of recognition. However, the syllables can have 2000 patterns and they are not very useful in languages like English, which does not possess a trivial syllabic division. In this case, *triphones* are more used, but their training is difficult (Young [14]).

This work proposes the use of phonemes as base for the Brazilian Portuguese speech recognition. The oral vowels (**a, é, i, ó, u, ê, ô**), were used in the recognition.

The energy coefficients of Wavelet Packet Transform with sub-bands, selected through the Mel scale, were chosen as features of the speech.

A new hierarchical Committee Machine decision system is presented. The classification of vowel signals is based on Support Vector Machines (SVM), where the base of decision is the tongue position and the rounding of the lips.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 159 – 166, 2006.

Section 2, presents the signal pre-processing phase. Section 3, shows the speech features extraction. Section 4, describes the training procedure of SVM neural network. Section 5, proposes a new technique for vowel recognition. Section 6 presents some experiments of vowels recognition.

2 Preprocessing

The preprocessing stage is composed of four steps: acquisition, filtering, pre-emphasis and normalization. In the acquisition step, the voice signal is sampled at a rate of 22050 Hz, with a bandwidth of 11050 Hz.

Signal frequencies above 10 kHz and electric power noise are eliminated through a band pass filter with cutoff frequencies of 80 Hz and 10 kHz. After that, the speech signal is pre-emphasized. In the normalization step, the maximum signal amplitude is normalized to one. Each frame is multiplied by a window function, named Hamming Window, in order to minimize any signal discontinuities in the time domain.

3 Features Extraction Using Wavelet Packets and Mel Scale

The Wavelet Packet (WP) decomposes the approximation spaces as well as details spaces, originating a binary tree structure. A WP decomposition facilitates the partitioning of the higher frequency side, of the frequency axis into smaller bands what cannot be achieved by using discrete wavelet transform [1].

The Mel scale is a signal representation scheme, used in the analysis of speech signals. Stevens and Volkmann in [12] defined the Mel scale as a frequency function of the magnitude of an auditory sensation. The Mel scale is linear in the frequency below 1000 Hz and logarithmic above this frequency.

Farroq and Datta in [4] had used Wavelet Packet with the Mel scale, which was found to be superior to Mel Frequency Cepstral Coefficients (MFCC) in unvoiced phoneme classification problem.

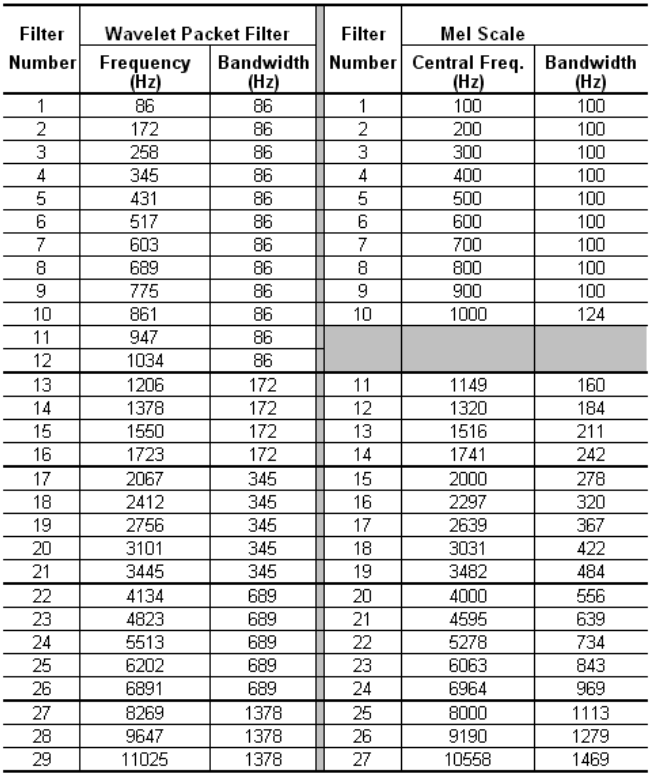
Gowdy and Tufekci in [5] evaluated the performance of the Wavelet Packet with Mel scale and compared its performance with MFCC coefficients. The results obtained through Wavelet Packet with Mel scale showed better recognition rates than MFCC for a phoneme recognition task.

In this work, seven levels of decomposition of the WP are utilized and the Mel scale is used to select 29 sub-bands.

First, a full seven level WP decomposition is carried out. Twelve subbands 86 Hz of the level 7, four subbands of 172 Hz of the level 6, five subbands of 345 Hz of the level 5, five subbands of 689 Hz of the level 4 and three subbands of 1378 Hz of the level 3 are utilized. The bandwidth obtained from each filter using WP decomposition is given in Table 1.

Therefore, the speech signal feature is represented by a vector whose 29 elements represent the energy of each sub-band extracted from the WP through the Mel scale. The used Wavelet mother was db5 (Daubechies [2]).

Table 1. Frequency bands achieved by Wavelet Packet decomposition and Mel scale



Filter	Wavelet Packet Filter		Filter	Mel Scale	
Number	Frequency (Hz)	Bandwidth (Hz)	Number	Central Freq. (Hz)	Bandwidth (Hz)
1	86	86	1	100	100
2	172	86	2	200	100
3	258	86	3	300	100
4	345	86	4	400	100
5	431	86	5	500	100
6	517	86	6	600	100
7	603	86	7	700	100
8	689	86	8	800	100
9	775	86	9	900	100
10	861	86	10	1000	124
11	947	86			
12	1034	86			
13	1206	172	11	1149	160
14	1378	172	12	1320	184
15	1550	172	13	1516	211
16	1723	172	14	1741	242
17	2067	345	15	2000	278
18	2412	345	16	2297	320
19	2756	345	17	2639	367
20	3101	345	18	3031	422
21	3445	345	19	3482	484
22	4134	689	20	4000	556
23	4823	689	21	4595	639
24	5513	689	22	5278	734
25	6202	689	23	6063	843
26	6891	689	24	6964	969
27	8269	1378	25	8000	1113
28	9647	1378	26	9190	1279
29	11025	1378	27	10558	1469

4 Training

In order to provide a better choice of the frames that represent the speech signal, instead of using all the frames, the signal was segmented using the Kmeans algorithm (Duda and Hart [3]) with two classes. Each signal frame possesses a vector characterized by 29-band energies selected by WP.

The Kmeans algorithm uses these vectors for signal separation. This procedure results in a significant reduction of the training time and an improvement of the performance of the system.

Figure 1, shows this procedure, where the frames were selected through Kmeans from a vowel "a" signal, using an energy vector.

It is perfectly clear that the frames were selected in the nearness of the center of the signal. This selection process avoids the use of frames that can represent variations of pronounce or noise, which generally occurs in the beginning and in the end of the location.

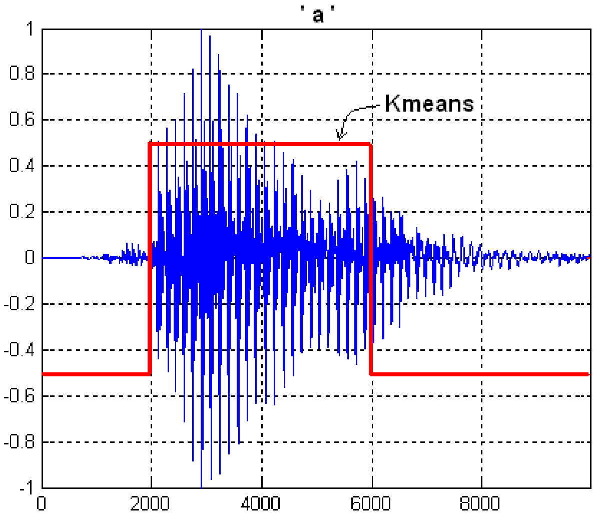


Fig. 1. Segmentation vowel '**a**' through Kmeans, using an energy vector

4.1 Support Vector Machine – SVM

Support Vector Machines (SVMs) represent a new approach for pattern classification, what has recently attracted a great interest in the machine learning community. Their appeal lies in their strong connection with the underlying statistical learning theory, in particular, the theory of Structural Risk Minimization.

The SVM theory was first introduced by Vapnik in [13]. The SVM learn the boundary regions between samples belonging to two classes, by mapping the input samples into a high dimensional space, and seeking a separating hyperplane in this space. The separating hyperplane is chosen in such a way that it maximizes its distance to the closest training samples.

Juneja [8], demonstrated the utility of the SVM in the classification of phonemes, resulting on a better performance than HMM (Hidden Markov Model).

Russell and Bilmes [9], affirm that in the last years it was verified a growing interest on classifiers that can go beyond the performance of the HMM.

To validate the use of SVM in the training stage, two experiments were carried out.

In the first test, the traditional strategy "one vs. all" was used in association with a decision scheme based on a Machine of Committee formed by a mixture of specialists (Haykin [6], pp. 402).

In second test, a new strategy, called Hierarchic Committee Machine-HCM, was used. This new strategy based on the articulatory phonetic is presented in Section 5.

5 Hierarchic Committee Machine – HCM

The proposed HCM is based on the characteristic vowel articulation of the Portuguese language. In phonetics, a *vowel* is a sound in spoken language, characterized by an open configuration of the vocal tract, without obstruction of air pressure above the glottis [11].

5.1 Articulatory Phonetic: Vowels Classification

The articulatory features that distinguish different vowels in a language are said to determine the vowel's *quality*. The vowels are described in terms of the common features: *height* (vertical tongue position), *backness* (horizontal tongue position) and *roundedness* (lip position), as shown in figure 2.

Height refers to the vertical position of the tongue relative to either the roof of the mouth or the aperture of the jaw. In high vowels, such as [i] and [u], the tongue is positioned high in the mouth, whereas in low vowels, such as [a], the tongue is positioned low in the mouth.

Backness refers to the horizontal tongue position during the articulation of a vowel relative to the back of the mouth. In front vowels, such as [i], the tongue is positioned forward in the mouth, whereas in back vowels, such as [u], the tongue is positioned towards the back of the mouth.

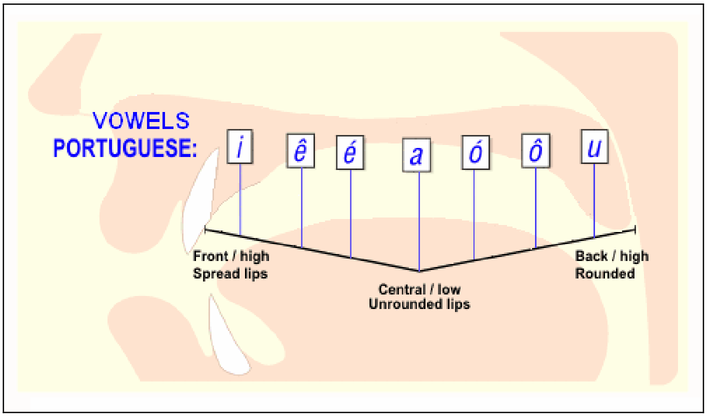


Fig. 2. Articulatory features: Height, Backness and Roundedness. Portuguese vowels.

Roundedness, refers to whether the lips are rounded or not. In most languages, roundedness is a reinforcing feature of mid to high back vowels, and not distinctive.

5.2 Hierarchical decision

Hosom in [7], used three neural networks specialists to detect the manner of articulation, place of articulation and height of the tongue in the production of phonemes. The outputs of the three neural networks were evaluated by a classifier using the Bayes rule. The obtained experimental results were better than those obtained through HMM.

Figure 3, shows the proposed new classification structure, in which characteristics like the tongue height and roundedness of the lips are the base for decision process.

The system is composed by seven SVM specialists, in which machine 01 selects phoneme /a/ through the strategy “one vs. all”.

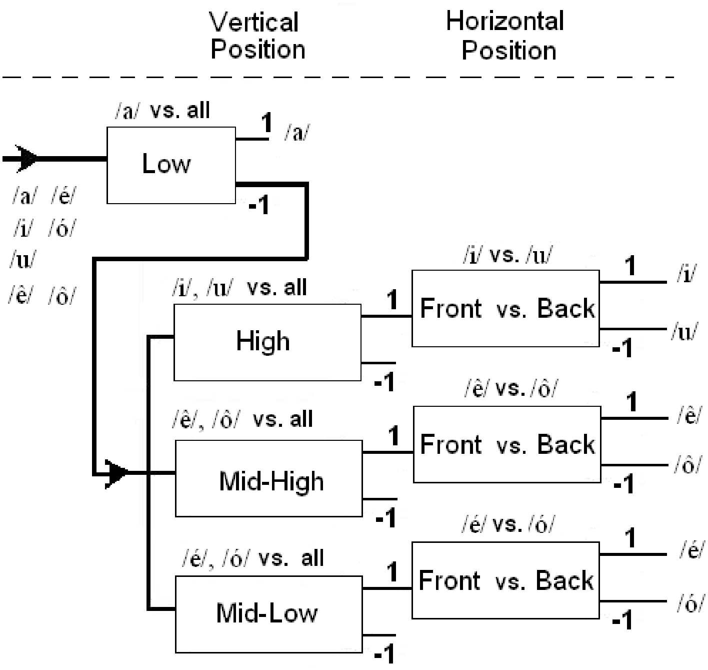


Fig. 3. New Classification system based on tongue position and roundedness of the lips

The phoneme /a/ is classified as *Central* and *Low*. Since vowel /a/ differs from the other vowels, its classification is made in first place.

In the next decision step, the system verifies if the pattern is High, Mid-high or Mid-low (Vertical Tongue Position). Having the biggest number of positive classifications, the specialist machine is declared winner. According to the winner, the system will classify the phoneme based on to the horizontal tongue position (Front vs. Back) in a strategy “one vs. one”.

The classification based on the roundedness of the lips is equivalent to that one based on the horizontal tongue position.

6 Experimental Results

In order to validate the proposed classification scheme, two experiments were performed: the first one, with the traditional strategy (one vs. all); the second, with the proposed hierarchical strategy.

For the user-dependent case, the training set was composed of 90 patterns. A set of 560 patterns was utilized for testing. The traditional strategy results showed success rates of 93.93%. The hierarchical strategy results showed success rates of 97.5%. Table 2, shows the confusion matrix for the hierarchical strategy in the user-dependent case.

Table 2. Confusion Matrix - Hierarchical strategy for user-dependent case

-	a	é	i	ó	u	ê	ô	% Success rate
a	80	0	0	0	0	0	0	100
é	0	80	0	0	0	0	0	100
i	0	0	80	0	0	0	0	100
ó	0	2	0	78	0	0	0	97.50
u	0	0	1	0	79	0	0	98.75
ê	0	0	1	0	4	75	0	93.75
ô	0	3	0	0	0	3	74	92.50

In the user-independent case, the training set was composed of 108 patterns. A set of 812 patterns was utilized for testing. The traditional strategy results show success rates of 87.44%. The hierarchical strategy results show success rates of 91.01%.

7 Conclusion

In this work, a new phoneme recognition system is proposed, where the tongue position and roundedness of the lips are adopted as base of decision. The coefficients of Wavelet Packet Transform with subbands selected through the Mel scale were selected as speech features. The Support Vector Machine was used as classifier in the structure of a Hierarchical Committee Machine. The database used for the recognition was a set of oral vowel phonemes of the Portuguese language.

The experimental results showed success rates of 97.50% for the user-dependent case and 91.01% for the user-independent case. This new proposal increased 3.5% the success rate in relation to the "one vs. all" decision strategy.

Therefore, we conclude that the new proposal presented better recognition taxes than the traditional strategy (one vs. all). Moreover, for the phonemes /a/, /é/ and /i/, the recognition rate was 100% for the user-dependent case.

The new hierarchical strategy decision proved to be more efficient, faster and robust, achieving a significant reduction in the complexity of the decision process.

Acknowledgments. This work was supported by CAPES (Federal government of Brazil).

References

1. Burrus, Sidney C. Gopinath, R. A. and Guo, Haitao.: Introduction to Wavelets and Wavelets Transforms. Prentice Hall, New Jersey. (1998).
2. Daubechies, I.: The Wavelet Transform, time-frequency localization and signal analysis. IEEE Trans. Inf. Theory, pp. 961-1005, (1990).
3. Duda R.O. and Hart, P.E.: Pattern classification and scene analysis. John Wiley & Sons, New York, (1973).
4. Farooq, O. and Datta, S.: Mel filter-like admissible wavelet packet structure for speech recognition. IEEE Signal Processing Letters. Vol. 08, Issue 07, pp. 196-198. July (2001).

5. Gowdy, J.N. and Tufekci Z.: Mel-scaled discrete wavelet coefficients for speech recognition. Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1351-1354. (2000).
6. Haykin, Simon.: Redes Neurais, Princípios e prática. 2ª Edição, Porto Alegre, Editora Bookman, (2001).
7. Hosom, John P.: Automatic Phoneme Alignment Based on Acoustic-Phonetic Modeling. International Conference on Spoken Language Processing-ICSLP'02, Boulder, Co., vol. I, pp 357-360, Sep. (2002).
8. Juneja, A. and Espy-Wilson, C.: Speech segmentation using probabilistic phonetic feature hierarchy and support vector machines. Proceedings of International Joint Conference on Neural Networks, Portland, Oregan, (2003).
9. Russell, Martin J. and Bilmes, Jeff A.: Introduction to the special issue on new computational paradigms for acoustic modeling in speech recognition. Editorial, Computer Speech and Language, nº 17, pp. 107-112, March (2003).
10. Santos, S. C. and Alcaim, Abraham.: Sílabas como unidades fonéticas para o reconhecimento de voz em português, *SBA Controle & Automação.* vol. 12, nº 01. (2001).
11. Silva, Thais Cristofáro.: Fonética e Fonologia do Português. 7º Edição, São Paulo, Ed. Contexto, (2003).
12. Stevens, S. S. Volkman, J. e Newman, E. B.: A Scale for Measurement of the Psychological Magnitude Picth. Journal of the Acoustical Society of America, vol. 08, pp. 185-190, January (1937).
13. Vapnik, V. N.: Principles of risk minimization for learning theory. Advances in Neural Information Processing Systems. vol. 04, pp.831-838, San Mateo, CA. (1992).
14. Young, S.: A Review of Large-Vocabulary Continuous-Speech Recognition, *IEEE Signal Processing Magazine,* pp. 45-57. Set. (1996).

A Passport Recognition and Face Verification Using Enhanced Fuzzy Neural Network and PCA Algorithm

Kwang-Baek Kim[1] and Sungshin Kim[2]

[1] Department of Computer Eng., Silla University, Busan 617-736, Korea
gbkim@silla.ac.kr
[2] School of Electrical and Computer Eng., Pusan National University, Busan, Korea
sskim@pusan.ac.kr

Abstract. In this paper, passport recognition and face verification methods which can automatically recognize passport codes and discriminate forgery passports to improve efficiency and systematic control of immigration management are proposed. Adjusting the slant is very important for recognition of characters and face verification since slanted passport images can bring various unwanted effects to the recognition of individual codes and faces. The angle adjustment can be conducted by using the slant of the straight and horizontal line that connects the center of thickness between left and right parts of the string. Extracting passport codes is done by Sobel operator, horizontal smearing, and 8-neighbornood contour tracking algorithm. The proposed RBF network is applied to the middle layer of RBF network by using the fuzzy logic connection operator and proposing the enhanced fuzzy ART algorithm that dynamically controls the vigilance parameter. After several tests using a forged passport and the passport with slanted images, the proposed method was proven to be effective in recognizing passport codes and verifying facial images.

1 Introduction

Because of globalization and the improvement of transportation, the number of people that arrive from and depart to different countries from airports has increased. The clerk of immigration control currently uses his/her bare eye to verify the passport. The purpose of immigration control is to find forgery, criminal, illegal immigrants, or someone prohibited from departing the country. A passport has information about the owner's identification photograph, nationality, name, social security number, gender, passport number, and so on.

It is difficult to use only bare eyes to distinguish and control the immigration process [1]. Time will be delayed, and due to obscure and unsure methods, accurate search of people who shouldn't be allowed in the country will not be possible. Therefore, this paper shows how to extract a string area of codes by applying Sobel operator, horizontal smearing, and 8-neighborhood contour tracking algorithm. The extracted string area becomes binary by applying a repeating binary

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 167–176, 2006.

method, which is applied with a CDM (Conditional Dilation Morphology) mask in order to recover the characters of an individual code [2],[3].

In order to extract individual codes from the string area to which CDM mask is applied, the individual code is extracted by 8-neighborhood contour tracking algorithm [4]. The remainder of this study is organized as follows. Section 2 presents the code extraction and slant compensation in detail. Passport recognition and forgery detection algorithms are introduced in Section 3 and 4, respectively. The experimental results are discussed in Section 5. Conclusions are drawn in Section 6.

2 Passport Code Extraction and Slant Compensation

The user information is represented in one code that is placed in the bottom of the passport. The passport code must be extracted in order to recognize the user information. In this paper, real passports that are currently in use are used to extract code areas that consist of 44 characters stands in two rows.

2.1 Code Extraction

The edge is detected by applying the Sobel mask to an original image of the passport, and horizontal smearing is applied to the image in which the Sobel mask has been applied. The method for extracting the string area of codes by applying the 8-neighborhood contour tracking algorithm to the horizontally smeared images is as following.

P_i^r and P_i^c are the vertical and horizontal pixels of the string areas of the extracted code, P_i^{r+1} and P_i^{c+1} are the next progressing vertical and horizontal pixels, respectively. P_s^r and P_s^c are vertical and horizontal pixels of the first contour tracking mask, respectively.

Step 1. Initialize with Eq. (1) in order to apply 8-neighbornood contour tracking algorithm to the string code area, and find the pixel by applying progressing mask as shown in Fig. 1.

$$P_i^{r-1} = P_i^r, \quad P_i^{c-1} = P_i^c \tag{1}$$

Step 2. When a black pixel is found after applying the progressing mask in the current pixel, calculate the value of $P_i^r$and P_i^c as shown in Eq. (2).

$$P_i^r = \sum_{i=0}^{7} P_i^{r+1}, \; P_i^c = \sum_{i=0}^{7} P_i^{c+1} \tag{2}$$

Step 3. For the 8 progressing masks, apply Eq. (3) to decide the next progressing mask.

$$\text{If } P_i^r = P_i^{r+1} \text{ and } P_i^c = P_i^{c+1} \text{ then rotates counter-clockwise} \tag{3}$$

Step 4. Stop if P_i^r and P_i^c return back to $P_s^r$and P_s^c or go back to the Step 1 and repeat. If $|P_i^r - P_s^r| \leq 1$ and $|P_i^c - P_s^c| \leq 1$ then Break, else go back to the Step 1.

6	5	4
7		3
0	1	2

EE

7	6	5
0		4
1	2	3

SE

0	7	6
1		5
2	3	4

SS

1	0	7
2		6
3	4	5

SW

2	1	0
3		7
4	5	6

WW

3	2	1
4		0
5	8	7

NW

4	3	2
5		1
6	7	0

NN

5	4	3
6		2
7	0	1

NE

Fig. 1. 8-neighborhood contour tracking process mask

2.2 Slant Compensation of Image

Since passport images can be tilted during the scan, “image slant compensation” is very important for face verification. If there is no slant during the extracting of strings of passport codes, extracting strings by selecting two areas that form maximum section by horizontal projection is possible. However, if slanting exists, this method is not useful. Skew compensation is applied by selecting the longer of two extracted strings, and then using the straight line that connects the center of the string’s thickness of the left and right sides and the slant of the horizontal line of that string. The extraction of code area and the image tilt compensation of the proposed method are shown in Fig. 2.

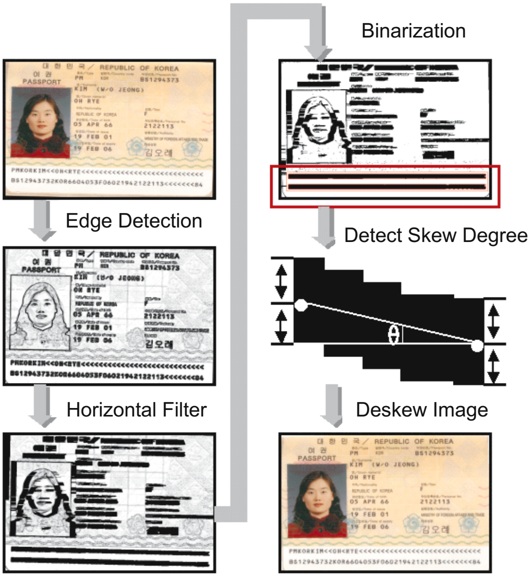


Fig. 2. Code character detection and skew compensation

2.3 Image Enhancement and Extraction of Individual Codes

CDM mask shown in Fig. 3 is used in order to transform the extracted string area to binary information, and to restore the characters of the individual code of the binarized string area.

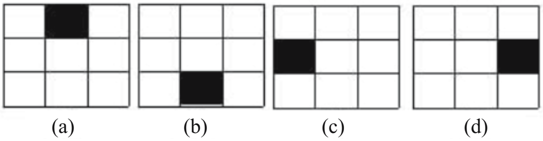


Fig. 3. CDM mask

The first step, Fig. 3(a), reconstructs bounding box's top horizontal outermost portion if the mask reach to character information into interior for horizontal direction by top-down method. The second step reconstructs left vertical elements by using a left-right method. The third step reconstructs horizontal elements of character from the bottom by using a bottom-up method. The fourth step reconstructs vertical elements of character from the right by using a right-left method.

Because the number of pixels that CDM mask is applied to vertical elements among the outermost pixels in a pixel per 3×3 mask, the image scanned in low restoration, 150 dpi, is available effectively. Fig. 4 shows the process that converges for up, down, right, and left directions in application form of CDM mask. After applying CDM Mask, 88 individual codes are extracted by using 8-neighborhood contour tracking algorithm. Fig. 5 shows the result of extracting individual codes with 8-neighborhood contour tracking algorithm.

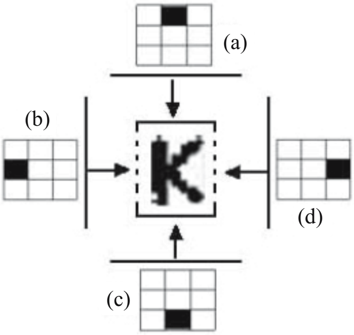
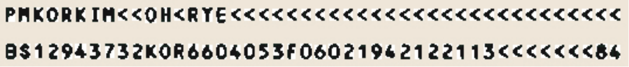


Fig. 4. Application of CDM mask

3 Passport Recognition by the Enhanced Fuzzy ART Based RBF Network

The RBF (Radial Basis Function) network based on enhanced fuzzy ART (Adaptive Resonance Theory) is applied for passport recognition. The proposed fuzzy



PMKORKIM<<OH<RYE<<<<<<<<<<<<<<<<<<<<<<<<<<<<

B$12943732KOR6604053F0602194212211 3<<<<<<<84

Fig. 5. Extraction result for individual character

RBF network is divided into two stages: a fuzzy logic connection operator to control vigilance parameters and Delta-bar-Delta to control the learning rate. Fuzzy ART is a self-learning algorithm that combines fuzzy logic and the ART learning model [5]. The vigilance parameter determines the allowable degree of mismatch between any input pattern and stored pattern [6]. Yager's intersection operator is defined as the following [7].

$$\mu(x_i) = 1 - Min\left[1,\ \{(1-X_1)^p + \ \cdots\ + (1-X_n)^p\}^{1/p}\right] \tag{4}$$

Let T^p and T^{p^*} be the target values of the learning pattern and the winner node, respectively. The equation to apply Yager's intersection operator and dynamically control vigilance parameters is as Eq. (5).

$$\begin{aligned} &\text{If} \quad T^p = T^{p^*} \ \text{then} \\ &\rho_{j^*}(t+1) = 1 - \wedge\left[1,\ \left\{(1-\rho_{j^*}(t))^2 + \left(1-\rho_{j^*}(t-1)\right)^2\right\}^{1/2}\right] \end{aligned} \tag{5}$$

The equation for controlling weight W from the conventional fuzzy ART algorithm is as following.

$$W(t+1) = \beta\,(X \wedge W(t)) + (1-\beta)\ W(t-1) \tag{6}$$

The recognition rate decreases if the value of β is too large in the conventional fuzzy ART [8]. Therefore, the learning parameter β is controlled dynamically as shown in Eq. (7) by considering actual distortion between stored patterns and learning patterns.

$$\beta = \frac{1}{1-\rho} \times \left(\frac{\|w_{j^*i} \wedge x_i\|}{\|x_i\|} - \rho\right) \tag{7}$$

Delta learning method is applied to update parameters between the middle and output layers. The output vector in the output layer can be calculated by Eq. (8), and normalized by the sigmoid function in Eq. (9).

$$O_k = \left(\sum_{j=1}^{M} w_{kj} \times O_j\right) \tag{8}$$

$$f(x) = \frac{1}{1+e^{-x}} \tag{9}$$

The equations to get the error value and error signal by comparing the normalized output vector and target vector is as following.

$$E = \frac{1}{2}(T_k^p - O_k)^2 \tag{10}$$

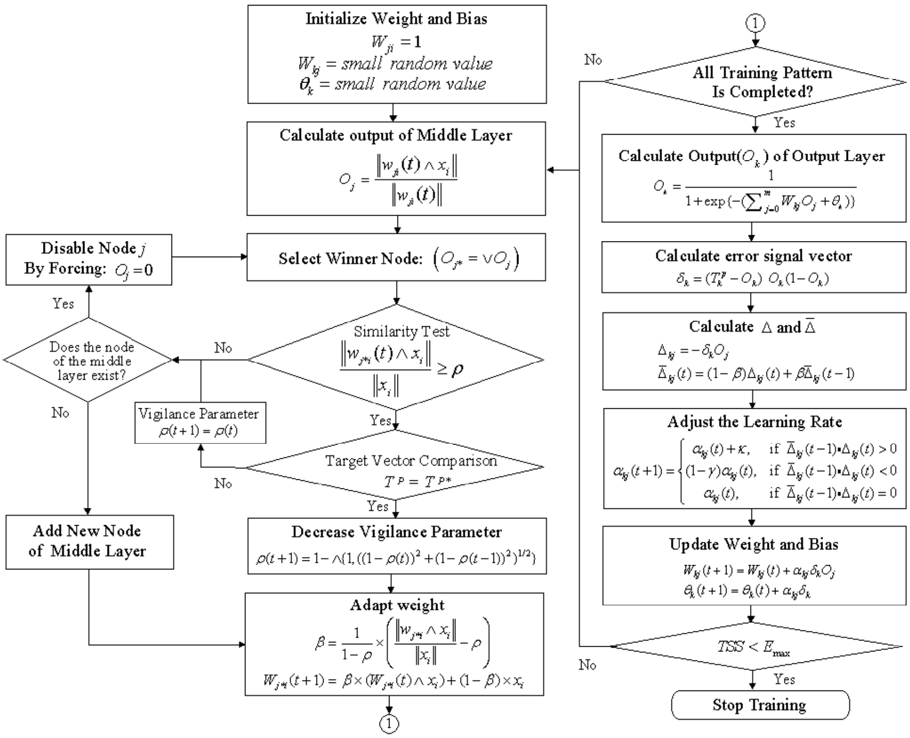

Initialize Weight and Bias
$W_{ji} = 1$
W_{kj} = small random value
θ_k = small random value
Calculate output of Middle Layer
$O_j = \frac{\|w_{ji}(t) \wedge x_i\|}{\|w_{ji}(t)\|}$
Disable Node j
By Forcing: $O_j = 0$
Select Winner Node: $(O_{j^*} = \vee O_j)$
Yes
Does the node of the middle layer exist?
No
Similarity Test
$\frac{\|w_{j^*i}(t) \wedge x_i\|}{\|x_i\|} \geq \rho$
No
Vigilance Parameter
$\rho(t+1) = \rho(t)$
Yes
Target Vector Comparison
$T^p = T^{p*}$
No
Yes
Add New Node of Middle Layer
Decrease Vigilance Parameter
$\rho(t+1) = 1 - \wedge\{1, ((1-\rho(t))^2 + (1-\rho(t-1))^2)^{1/2}\}$
Adapt weight
$\beta = \frac{1}{1-\rho} \times \left(\frac{\|w_{j^*i} \wedge x_i\|}{\|x_i\|} - \rho\right)$
$W_{j^*i}(t+1) = \beta \times (W_{j^*i}(t) \wedge x_i) + (1-\beta) \times x_i$
1
1
No
All Training Pattern Is Completed?
Yes
Calculate Output(O_k) of Output Layer
$O_k = \frac{1}{1+\exp\{-(\sum_{j=0}^{m} W_{kj} O_j + \theta_k)\}}$
Calculate error signal vector
$\delta_k = (T_k^p - O_k)\, O_k(1-O_k)$
Calculate Δ and $\overline{\Delta}$
$\Delta_{kj} = -\delta_k O_j$
$\overline{\Delta}_{kj}(t) = (1-\beta)\Delta_{kj}(t) + \beta\overline{\Delta}_{kj}(t-1)$
Adjust the Learning Rate
$\alpha_{kj}(t+1) = \begin{cases} \alpha_{kj}(t) + \kappa, & \text{if } \overline{\Delta}_{kj}(t-1) \cdot \Delta_{kj}(t) > 0 \\ (1-\gamma)\alpha_{kj}(t), & \text{if } \overline{\Delta}_{kj}(t-1) \cdot \Delta_{kj}(t) < 0 \\ \alpha_{kj}(t), & \text{if } \overline{\Delta}_{kj}(t-1) \cdot \Delta_{kj}(t) = 0 \end{cases}$
Update Weight and Bias
$W_{kj}(t+1) = W_{kj}(t) + \alpha_{kj}\delta_k O_j$
$\theta_k(t+1) = \theta_k(t) + \alpha_{kj}\delta_k$
$TSS < E_{\max}$
No
Yes
Stop Training

Fig. 6. Schematic diagram of the RBF network based on enhanced fuzzy ART

$$\delta_k = (T_k^p - O_k)\, O_k(1 - O_k) \tag{11}$$

After getting Delta-bar-Delta using Eq. (12) for the dynamic adjustment of the learning rate, the learning rate is dynamically adjusted by Eq. (13).

$$\begin{aligned} \Delta_{kj} &= -\delta_k O_j \\ \overline{\Delta}_{kj} &= (1-\beta)\Delta_{kj}(t) + \beta\overline{\Delta}_{kj}(t-1) \end{aligned} \tag{12}$$

$$\alpha_{kj}(t+1) = \begin{cases} \alpha_{kj}(t) + \kappa, & \text{if } \overline{\Delta}_{kj}(t-1) \cdot \Delta_{kj}(t) > 0 \\ (1-\gamma)\alpha_{kj}(t), & \text{if } \overline{\Delta}_{kj}(t-1) \cdot \Delta_{kj}(t) < 0 \\ \alpha_{kj}(t), & \text{if } \overline{\Delta}_{kj}(t-1) \cdot \Delta_{kj}(t) = 0 \end{cases} \tag{13}$$

The equations for weight and bias are updated as in Eq. (14) and Eq. (15). The schematic diagram of the proposed RBF network based on enhanced fuzzy ART is shown in Fig. 6.

$$w_{kj}(t+1) = w_{kj}(t) + \alpha_{kj}\delta_k O_j \tag{14}$$

$$\theta_k(t+1) = \theta_k(t) + \alpha_{kj}\delta_k \tag{15}$$

4 Forgery Detection by Face Verification

The recognized passport code information is used to obtain the feature vectors of facial image that is acquired by PCA algorithm from the database.

4.1 PCA

PCA finds the collection of certain normalized orthogonal axis that indicates to each direction of maximum covariance for input data. The learning method using PCA is as following [9]. The two-dimensional image can be presented by a vector, and the k number of learned image vectors can be presented by $X = \left[x^1|x^2|x^3|\cdots|x^k\right]$'s rows. An image's average vector can be acquired by Eq. (16) and the difference between the one-dimensional image vector and average image vector can be acquired by Eq. (17).

$$m = \frac{1}{k}\sum_{j=1}^{k} x^i \tag{16}$$

$$\overline{x}^i = x^i - m \tag{17}$$

By using the k number of $\overline{x}^i$ vectors which is the result of Eq. (17), the $\overline{X} = \left[\overline{x}^1|\overline{x}^2|\overline{x}^3|\cdots|\overline{x}^k\right]$ row can be acquired. $\overline{X}$ row can be used to obtain the covariance matrix by using Eq. (18).

$$\Omega = \overline{X}\,\overline{X}^T \tag{18}$$

The method for representing the studied images in PCA data is as following. After obtaining the $V = \left[v^1|v^2|v^3|\cdots|v^k\right]$ by using eigenvectors that are acquired through the covariance matrix, obtain the property vectors of the studied images using Eq. (19).

$$\tilde{x}^i = V^T\bar{x}^i \tag{19}$$

For face recognition using PCA, first the target image is subtracted from the average image to get the $\overline{y}^i$ image. Eq. (20) shows how the $\overline{y}^i$ image is acquired. Then, using the transposed matrix of the eigenvector, the feature vector of the target image is obtained as in Eq. (21).

$$\overline{y}^i = y^i - m \tag{20}$$

$$\widetilde{y}^i = V^T\overline{y}^i \tag{21}$$

4.2 Extraction of Facial Area of Passport Picture

The position of the picture in the passport is in between 1/5 and 4/5 of the vertical length and 1/3 of the horizontal width of the passport image based on the top left of the extracted code string. From the center of 2/3 of the width in the candidate area, we extract 50 pixels of the width, 130 pixels of the length, from the left and right. The final extracted region is used the face area. The method for extracting a passport picture is shown in Fig. 7.

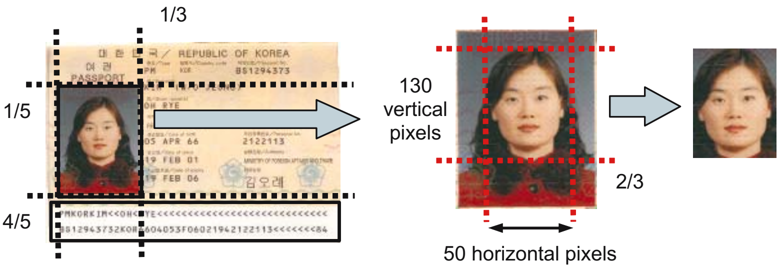


Fig. 7. Face area detection of passport picture

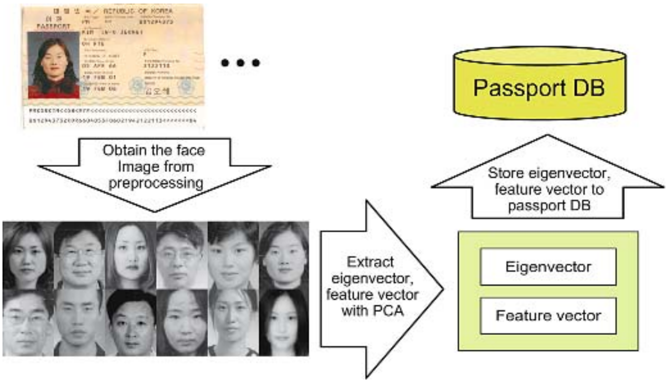


Fig. 8. Database construction for face information

4.3 Database Construction for Face Information

First, acquire images of facial area by the process of extracting facial areas from several passports. Study the acquired facial images using the PCA algorithm, and add the unique vector and feature vector of the learned facial images to the database. By using the information of the unique vector and feature vector, the verification of facial similarity is possible. The process of database construction of facial information is shown in Fig. 8.

4.4 Face Authentication

After acquiring the unique vector and feature vector from both the database and actual passport, face authentication of a passport can be done by calculating the feature vector of facial images by using Eq. (17) and (18).

The similarity of feature vectors between the calculated facial image and the database can be calculated using Eq. (19). If the similarity rate exceeds a certain critical value, the passport is valid; if not, it is possible to assume that the passport is forged.

Table 1. Nodes of created middle layer

Learning Algorithm	Pattern	Nodes of created middle layer
RBF network based enhanced fuzzy ART	Numeric	85
	Character	162

Table 2. Recognition rate of passport

	Character	Numeric	Recognition rate
Normal	1045/1116	2034/2052	97%
Slant compensation	1116/1116	2052/2052	100%

5 Analysis of Experiment and Result

The experiment was conducted by VC++ 6.0 on an Intel Pentium-IV 2GHz CPU. Twelve 600×437 sized images from the passport which are scanned by HP ScanJet 4200C, twelve images with facial forgery, and twelve images with fake picture were used for this experiment.

The result of extracting individual characters from a passport image is shown in Fig. 5. The 72 string areas from 36 passport images are all extracted, and both 2052 individual code characters and 1116 individual numbers are all extracted. 100 number codes and 260 character codes among the extracted 3168 passport codes are trained by applying the enhanced fuzzy ART based RBF Network algorithm. The parameter setting of the enhanced fuzzy ART based RBF network is as follows: $\alpha(0.7)$ is learning rate, $\mu(0.9)$ is momentum, and $\kappa(0.00005)$, $\gamma(0.001)$, $\beta(0.9)$ are delta-bar-delta constants. The number of nodes in the middle layer to which the enhanced fuzzy ART based RBF Network algorithm is applied is shown in Table 1.

The recognition rate of the 36 passport images made for the efficiency test is shown in Table 2. The passport images are recognized 97% of the time by mere scanning, but they are recognized 100% of the time by scanning using image-slant compensation.

12 original passports, 12 passports with fake pictures, and 12 passports with forged facial areas were used. The facial verification similarity was set at 0.8 for the experiment. The result is shown in Table 3. The passports with fake pictures and forged facial areas are distinguished as counterfeit. On the other hand, the original passports passed safely. Therefore, PCA algorithm is proven to be effective for face verification.

Table 3. Image verification of passport face

	Original passport	Face forgery	Picture forgery
Detection of forgery	0/12	12/12	12/12
Pass	12/12	0/12	0/12

6 Conclusion

The string codes are restored by applying CDM mask to the binary string area, and individual codes are extracted by 8-neighborhood contour tracking algorithm. The enhanced fuzzy ART based RBF network is applied to prevent different patterns from being classified as the same cluster or same patterns from being classified as different clusters. All twenty-four forged passports are detected during the face verification experiment by PCA algorithm that measures the similarity of facial feature vector. The experimental results show that the proposed facial recognition algorithm is effective.

References

1. Kim, K. B.: Intelligent Immigration Control System by Using Passport Recognition and Face Verification. In: Jun, W., Xiaofeng L., Zhang, Y. (eds.): Advances in Neural Networks-ISNN 2005. Lecture Notes in Computer Science. LNCS 3497. Springer-Verlag Berlin Heidelberg New York (2005) 147–156
2. Gonzalez, R. C. and Wintz, P.: Digital Image Processing. Addison-Wesley Publishing Company Inc. (1977)
3. Kim, K. B., Kim, S. S., Ha, S. A.: Recognition of Passports Using a Hybrid Intelligent System. In: Mohamed, K., Aurelio, C. (eds.): Image Analysis and Recognition. Lecture Notes in Computer Science. LNCS 3656. Springer-Verlag Berlin Heidelberg New York (2005) 540–548
4. Kim, K. B., Lim, E. K., Kim, G. H.: Analysis System of Endoscopic Image of Early Gastric Cancer. Journal of Fuzzy Logic and Intelligent Systems. Vol.15, No.4 (2005) 473–478
5. Carpenter, G. A., Grossberg, S., Rosen, D. B.: Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks, Vol.4, 1991 (759–771)
6. Kim, K. B., Kim, Y. J.: Recognition of English Calling Cards by Using Enhanced Fuzzy Radial Basis Function Neural Networks. IEICE Transactions on Fundamentals of Electronics, Communications and
Computer Sciences. Vol.E87-A, No.6 (2004) 1355–1362
7. Yager, R. P.: On a General Class of Fuzzy Connective. Fuzzy Sets Systems. Vol.4. 1980 (235–242)
8. Kim, K. B. : Recognition of Identifiers from Shipping Container Images Using Fuzzy Binarization and Enhanced Fuzzy Neural Network. In: Lipo, W., Yaochu, J. (eds.): Fuzzy Systems and Knowledge Discovery. Lecture Notes in Artificial Intelligence. LNAI 3613. 2005 (761–771)
9. Martinez, A. M., Kak, A. C.: PCA versus LDA. IEEE Trans. Pattern Analysis and Machine Intelligence. Vol.23, Issue.2, 2001 (228–233)

A Weighted FMM Neural Network and Its Application to Face Detection*

Ho-Joon Kim[1], Juho Lee[2], and Hyun-Seung Yang[2]

[1] School of Computer Science and Electronic Engineering
Handong University, Pohang, 791-708, Korea
hjkim@handong.edu
[2] Department of Computer Science, KAIST
Daejeon, 305-701, Korea
{jhlee, hsyang}@paradise.kaist.ac.kr

Abstract. In this paper, we introduce a modified fuzzy min-max(FMM) neural network model for pattern classification, and present a real-time face detection method using the proposed model. The learning process of the FMM model consists of three sub-processes: hyperbox creation, expansion and contraction processes. During the learning process, the feature distribution and frequency data are utilized to compensate the hyperbox distortion which may be caused by eliminating the overlapping area of hyperboxes in the contraction process. We present a multi-stage face detection method which is composed of two stages: feature extraction stage and classification stage. The feature extraction module employs a convolutional neural network (CNN) with a Gabor transform layer to extract successively larger features in a hierarchical set of layers. The proposed FMM model is used for the pattern classification stage. Moreover, the model is utilized to select effective feature sets for the skin-color filter of the system.

1 Introduction

Fuzzy min-max (FMM) neural networks were introduced by Simpson [1] using the concept of hyperbox fuzzy sets. A hyperbox defines a region of the n-dimensional pattern space that has patterns with full class membership using its minimum point and its maximum point. The fuzzy min-max neural networks are built by making one pass through the input patterns and forming hyperboxes into fuzzy sets to represent pattern classes. Gabrys and Bargiela have proposed a General Fuzzy Min-Max (GFMM) neural network which is a generalization and extension of the FMM clustering and classification algorithm [2]. In GFMM method, input patterns can be fuzzy hyperboxes or crisp points in the pattern space. We present a modified FMM, called the weighted FMM (WFMM) neural network that takes the weights into account. The

* This research was supported as a 21st Century Frontier R&D Program and Brain Neuroinformatics Research Program sponsored by Ministry of Information and Communication and Minister of Commerce, Industry and Energy in KOREA.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 177 – 186, 2006.

rationale for this idea is that a feature of a particular hyperbox can cover many more training patterns than other features of the same hyperbox and features of other hyperboxes. A weight value is assigned to each of the dimensions of each hyperbox so that membership can be assigned considering not only the occurrence of patterns but also the frequency of the occurrences within that dimension.

Growing interest in computer vision has motivated a recent surge in research on problems such as face recognition, pose estimation, face tracking and gesture recognition. However, most methods assume human faces in their input images have been detected and localized [3-5]. Color usually presents a strong intuitive cue in complex scene images. Recently, skin detection has emerged as an active research topic in several practical applications including face detection and tracking. Various generic skin models in a number of color spaces have been presented [6-7]. However, we can expect variations when images are taken in various settings, with different kinds of camera hardware, and under a wide range of lighting conditions [8]. Therefore the generic skin model may be inadequate to accurately capture the wide distribution of skin colors in an individual image. In this paper we present a FMM-based feature analysis technique for the face detection system. Two kinds of relevance factors are defined to analyze the relationships between features and pattern classes. Through the feature analysis, we can select the most relevant features for the skin-color filter as well as the pattern classifier. Moreover, the training process can make it possible to adaptively adjust the feature ranges of the skin-color filter.

2 A Weighted FMM Neural Network

In our previous work, a weighted fuzzy min-max (WFMM) neural network has been introduced [9]. The model employs a new activation function which has the weight value for each feature in a hyperbox. In this paper, we introduce an improved structure of the WFMM neural network and its application to a face detection problem.

2.1 Structure and Behavior

The weighted fuzzy min-max(WFMM) neural network is a modified version of Simpson's FMM model[1]. The model consists of three layers: input layer, hyperbox layer and class layer. In the model, the membership function of a hyperbox is defined as Equation (1).

$$B_j = \{X, U_j, V_j, C_j, F_j, f(X, U_j, V_j, C_j, F_j)\} \ \forall X \in I^n \tag{1}$$

In the equation, U_j and V_j mean the vectors of the minimum and maximum values of hyperbox *j*, respectively. C_j is a set of the mean points for the feature values and F_j means the a set of frequency of feature occurrences within a hyperbox. As shown in Equation (2) and (3), the model employs a new activation function which has the factors of feature value distribution and the weight value for each feature in a hyperbox.

$$b_j(A_h) = \frac{1}{\sum_{i=1}^{n} w_{ji}} \bullet \sum_{i=1}^{n} w_{ji}[\max(0, 1-\max(0, \gamma_{jiv} \min(1, a_{hi} - v_{ji}))) \\ + \max(0, 1-\max(0, \gamma_{jiu} \min(1, u_{ji} - a_{hi}))) - 1.0] \tag{2}$$

$$\begin{cases} \gamma_{jiU} = \dfrac{\gamma}{R_U} & R_U = \max(s, u_{ji}{}^{new} - u_{ji}{}^{old}) \\ \gamma_{jiV} = \dfrac{\gamma}{R_V} & R_V = \max(s, v_{ji}{}^{old} - v_{ji}{}^{new}) \end{cases} \tag{3}$$

The hyperbox membership function has weight factor to consider the relevance of each feature as different values. In the equation, the w_{ij} is the connection weight between *i*-th feature and *j*-th hyperbox. The weighted FMM neural network is capable of utilizing the feature distribution and the weight factor in learning process as well as in classification process. Since the weight factor effectively reflects the relationship between feature range and its distribution, the system can prevent undesirable performance degradation which may be caused by noisy patterns. Consequently the proposed model can provide more robust performance of pattern classification when the training data set in a given problem includes some noise patterns or unusual patterns.

2.2 Learning Algorithm

The learning process of the model consists of three subprocesses: hyperbox creation, expansion, and contraction processes.

If the expansion criterion shown in Equation (4) has been met for hyperbox B_j, f_{ji}, u_{ji}, v_{ji} and c_{ij} are adjusted using Equation (5) and (6).

$$n\theta \geq \sum_{i=1}^{n} (\max(v_{ji}, x_{hi}) - \min(u_{ji}, x_{hi})) \tag{4}$$

$$\begin{cases} f_{ji}^{new} & = & f_{ji}^{old} + 1 \\ u_{ji}^{new} & = & \min(u_{ji}^{old}, x_{ki}) \\ v_{ji}^{new} & = & \min(v_{ji}^{old}, x_{ki}) \end{cases} \tag{5}$$

$$c_{ji}^{new} = (c_{ji} * f_{ji}^{old} + x_{hi}) / f_{ji}^{new} \tag{6}$$

As shown in the equations, the frequency value is increased by 1 at every expansion and the feature range expansion operation is similar to the fuzzy intersection and fuzzy union operations [6]. The mean point value, c_{ji}, is updated by Equation (6). During the learning process the weight values are determined by Equation (7) and (8).

$$w_{ji} = \frac{\alpha f_{ji}}{R} \tag{7}$$

$$R = \max\left(s, v_{ji} - u_{ji}\right) \tag{8}$$

As shown in the equations, the weight value is increased in proportion to the frequency of the feature. In the equations, s is a positive constant to prevent the weight from having too high value when the feature range is too small. The value of f_{ji} is adjusted through the learning process. The contraction process is considered as an optional part of our model. The contraction process is to eliminate the possible overlappings between hyperboxes that represent different classes. We can expect that the weights concept of the model replace the role of overlapping handling because the weights reflect the relevance of feature values and hyperbox as different values. We define a new contraction method including the weight updating scheme. To determine whether or not the expansion has created any overlapping, a dimension by dimension comparison between hyperboxes is performed. If one of the following four cases is satisfied, then overlapping exists between the two hyperboxes.

$$\begin{aligned}
&case1: u_{ji} < u_{ki} < v_{ji} < v_{ki}\\
&\qquad \delta^{new} = \min(v_{ji} - u_{ki}, \delta^{old})\\
&case2: u_{ki} < u_{ji} < v_{ki} < v_{ji}\\
&\qquad \delta^{new} = \min(v_{ki} - u_{ji}, \delta^{old})\\
&case3: u_{ji} < u_{ki} < v_{ki} < v_{ji}\\
&\qquad \delta^{new} = \min(\min(v_{ki} - u_{ji}, v_{ji} - u_{ki}), \delta^{old})\\
&case4: u_{ki} < u_{ji} < v_{ji} < v_{ki}\\
&\qquad \delta^{new} = \min(\min(v_{ji} - u_{ki}, v_{ki} - u_{ji}), \delta^{old})
\end{aligned}$$

For each of these cases, contraction process is performed. If $\delta^{old} - \delta^{new} > 0$, then $\Delta = i,\ \delta^{old} = \delta^{new}$, signifying that there was an overlap for Δ th dimension. Otherwise, the testing is terminated and the minimum overlap index variable is set to indicate that the next contraction step is not necessary, i.e. $\Delta = -1$. If $\Delta > 0$, then the Δ th dimension of the two hyperboxes are adjusted. Only one of the n dimensions is adjusted in each of the hyperboxes to keep the hyperbox size as large as possible and minimally impact the shape of the hyperboxes being formed.

As illustrated in Equation (9), we have defined new adjustment schemes from the new definition of hyperbox for the four cases. The frequency values, the mean points and the feature ranges are updated for the four cases. Consequently the frequency factor is increased in proportion to the relative size of the feature range, and the mean point value is adjusted by considering the expanded feature range.

$$case1: u_{j\Delta} < u_{k\Delta} < v_{j\Delta} < v_{k\Delta}$$

$$\begin{cases} v_{j\Delta}{}^{new} &= & v_{j\Delta}{}^{old} - \dfrac{f_{k\Delta}}{f_{j\Delta}+f_{k\Delta}}(v_{j\Delta}{}^{old} - u_{k\Delta}{}^{old}) \\ u_{k\Delta}{}^{new} &= & u_{k\Delta}{}^{old} + \dfrac{f_{j\Delta}}{f_{j\Delta}+f_{k\Delta}}(v_{j\Delta}{}^{old} - u_{k\Delta}{}^{old}) \\ f_{j\Delta}{}^{new} &= & f_{j\Delta}{}^{old} * (\dfrac{v_{j\Delta}{}^{new} - u_{j\Delta}{}^{new}}{v_{j\Delta}{}^{old} - u_{j\Delta}{}^{old}}) \\ f_{k\Delta}{}^{new} &= & f_{k\Delta}{}^{old} * (\dfrac{v_{k\Delta}{}^{new} - u_{k\Delta}{}^{new}}{v_{k\Delta}{}^{old} - u_{k\Delta}{}^{old}}) \\ c_{j\Delta}{}^{new} &= & u_{j\Delta}{}^{new} + (c_{j\Delta}{}^{old} - u_{j\Delta}{}^{old}) * (\dfrac{v_{j\Delta}{}^{new} - u_{j\Delta}{}^{new}}{v_{j\Delta}{}^{old} - u_{j\Delta}{}^{old}}) \\ c_{k\Delta}{}^{new} &= & u_{k\Delta}{}^{new} + (c_{k\Delta}{}^{old} - u_{k\Delta}{}^{old}) * (\dfrac{v_{k\Delta}{}^{new} - u_{k\Delta}{}^{new}}{v_{k\Delta}{}^{old} - u_{k\Delta}{}^{old}}) \end{cases} \tag{9}$$

3 A Face Detection Method Using the WFMM Model

As shown in Fig.1, our face detection system consists of three modules: preprocessor, feature extractor and pattern classifier. Through the skin color analysis and training process, the system can generate an adaptive skin model and a relevant feature set for the given illumination condition. The feature extractor generates numerous features from the input image. The number of features and the relevance factors of the features affect the computation time and the performance of the system. Therefore we propose a feature analysis technique to reduce the amount of features for the pattern classifier.

3.1 WFMM-Based Feature Analysis Technique

This section describes a feature analysis technique for the skin-color filter and the classifier. We define two kinds of relevance factors using the proposed FMM model as follows:

$RF1(x_j, C_k)$: *the relevance factor between a feature value* x_j *and a class* C_k

$RF2(X_i, C_k)$: *the relevance factor between a feature type* X_i *and a class* C_k

The first measure *RF1* is defined as Equation (9). In the equation, constant N_B and N_k are the total number of hyperboxes and the number of hyperboxes that belong to class k, respectively. Therefore if the $RF1(x_i, k)$ has a positive value, it means an excitatory relationship between the feature x_i and the class k. But a negative value of $RF1(x_i, k)$ means an inhibitory relationship between them. A list of interesting features for a given class can be extracted using the *RF1* for each feature.

$$RF1(x_i, C_k) = \left(\frac{1}{N_k} \sum_{B_j \in C_k} S(x_i, (u_{ji}, v_{ji})) \cdot w_{ij} - \frac{1}{(N_B - N_k)} \sum_{B_j \notin C_k} S(x_i, (u_{ji}, v_{ji})) \cdot w_{ij}\right) / \sum_{B_j \in C_k} w_{ij} \tag{9}$$

In Equation (9), the feature value x_i can be defined as a fuzzy interval which consists of min and max values on the *i*-th dimension out of the n-dimension feature space. The function ***S*** a similarity measure between two fuzzy intervals.

The second measure *RF2* can be defined in terms of *RF1* as shown in Equation (10). In the equation, L_i is the number of feature values which belong to *i*-th feature.

$$RF2(X_i, C_k) = \frac{1}{L_i} \sum_{x_l \in X_i} RF1(x_l, C_k) \tag{10}$$

The *RF2* shown in Equation (10) represents the degree of importance of a feature in classifying a given class. Therefore it can be utilized to select a more relevance feature set for skin color filter.

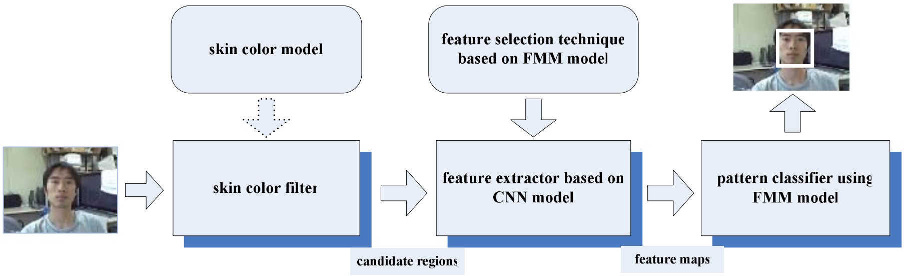

skin color model
feature selection technique based on FMM model
skin color filter
feature extractor based on CNN model
pattern classifier using FMM model
candidate regions
feature maps

Fig. 1. The face detection system using hybrid neural networks

3.2 Feature Extraction and Face Classification

The most advantageous feature of convolutional neural network is invariant detection capability for distorted patterns in images [2-3]. The underlying system employs a convolutional neural network in which a Gabor transform layer is added at the first layer. As shown in Fig. 2, the first layer of the network extracts local feature maps from the input image by using Gabor transform filters.

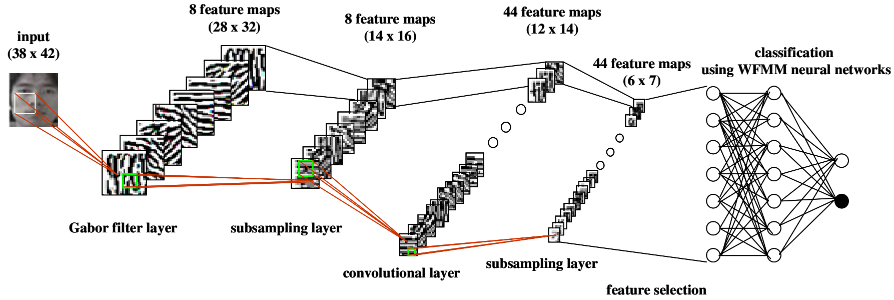


Fig. 2. Face detector using hybrid neural networks

The other layers of the feature extractor include two types of sub-layers called *convolution layer* and *sub-sampling layer*. Each layer of the network extracts successively larger features in a hierarchical set of layers. Finally a feature set is generated for the input of the pattern classifier. The number of the features can be reduced by the feature analysis technique using the FMM model described in the previous section. For the feature extractor, a set of (38 x 42) candidate areas are selected as input images. The first layer of the feature extractor, Gabor filter layer, extracts eight feature maps in which the size of feature map is (28 x 32). Each unit in each feature map is connected to a 11 x 11 neighborhood into the input retina. In the subsampling layer, the feature map has half the number of rows and columns of the input data. Therefore the layer has eight feature maps of size 14 x 16. The convolutional layer generates 44 feature maps. Each unit is connected to 3 x 3 neighborhood at identical locations in a subset of the feature maps of the Gabor transform layer. 1,848 feature values are generated and inputted into the input layer of the WFMM-based classifier. The aforementioned feature analysis technique can be used to reduce the number of these features.

4 Experimental Results

Two types of experiments have been conducted for a set of real images. For the training of skin-color filter, the system considers eleven color features, *Red, Green, Blue, Intensity, Cb, Cr, Magenta, Cyan, Yelleow, Hue and Saturation*. Fig. 3 shows two input images captured under different illumination conditions. Table 1 shows the skin-color analysis result and the feature range data derived from the training process. As shown in the table, different kinds of features can be adaptively selected for a given condition, and the feature ranges of skin-color filter can be also adjusted by the training process.

Table 1 shows four features which have the highest value of the relevance factor *RF1*. As shown in the table, a number of hyperboxes for face and non-face patterns have been generated and the relevance factors are also adjusted through the training process. Therefore the system can select more effective feature set adaptively for the given environment.

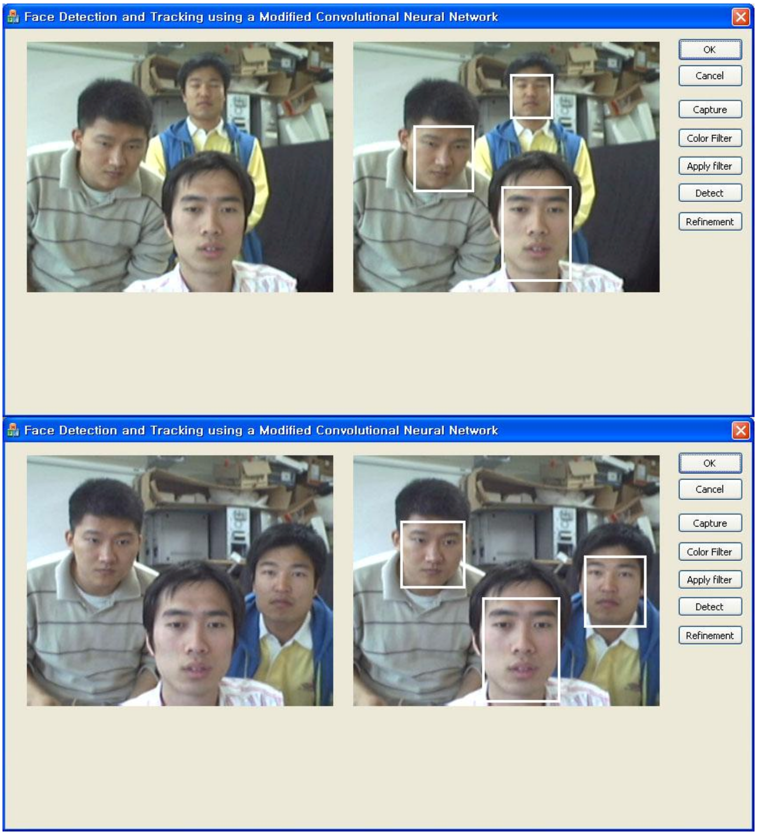

Face Detection and Tracking using a Modified Convolutional Neural Network
OK
Cancel
Capture
Color Filter
Apply filter
Detect
Refinement
Face Detection and Tracking using a Modified Convolutional Neural Network
OK
Cancel
Capture
Color Filter
Apply filter
Detect
Refinement

Fig. 3. Two training data captured under different illumination conditions

Table 1. Feature analysis results for the two images

image - 1			*image - 2*		
feature	feature range	*RF1*	feature	feature range	*RF1*
Hue	0.833 ~ 0.992	9.1103	*Cb*	0.589 ~ 0.772	0.8888
Saturation	0.019 ~ 0.135	9.0104	*Yellow*	0.433 ~ 0.632	0.7832
Cb	0.761 ~ 0.964	8.8212	*Saturation*	0.056 ~ 0.243	0.7204
Cr	0.053 ~ 0.234	6.7320	*Blue*	0.437 ~ 0.627	0.6929

We have selected face patterns from the real images and non-face patterns from the background images. 100 face patterns and 100 non-face patterns have used for training process. Fig. 4 shows the change of detection rate and false alarm rate by varying the number of training patterns, respectively. The result shows that the detection rate increases as more training patterns are used, and the false alarm rate decreases as more non-face counter examples are used for training.

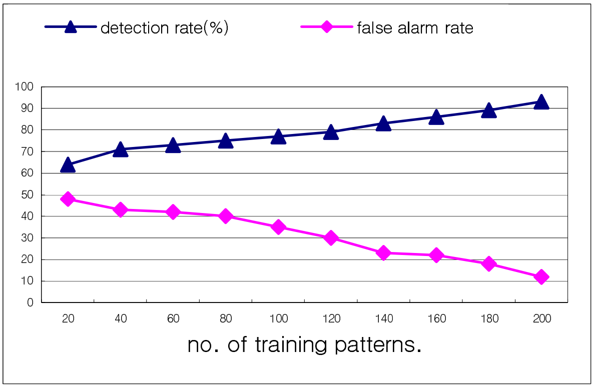


Fig. 4. Detection rate and false alarm rate as varying the number of training patterns

5 Conclusion

The proposed WFMM can provide at least two advantages over the original FMM: (a) it would work better for pattern classification than the original scheme, especially for data sets with highly uneven distribution of features or noisy features since the hyperboxes in the WFMM model is not too sensitive to a few occurrence of unusual/noisy features in input patterns, (b) the learned weights for each feature during training process can be used to identify the relevance of the feature to the given class, which can be easily used for possible rule generation. The feature relevance measures computed through the feature analysis can be also utilized in designing an optimal structure of the classifier. We have applied the proposed model to a real-time face detection system in which the illumination conditions are frequently changed.

References

1. Simpson, P. K.: Fuzzy Min-Max Neural Networks Part 1: Classification. IEEE Transaction on Neural Networks, Vol.3. No.5. (1997) 776-786
2. Gabrys, B. and Bargiela A.: General Fuzzy Min-Max Neural Network for Clustering and Classification. IEEE Transaction on Neural Networks, Vol.11. No.3. (2000) 769-783
3. Feraud, R., Bernier, O. J., Viallet J. E. and Collobert, M.: A Fast and Accurate Face Detector Based on Neural Networks, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.23, No.1.(2001) 42-53.
4. Garcia, Cristophe and Delakis, Manolis: Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.26, No.11, (2004) 1408-1423
5. Lawrence, Steve, Giles, C. L., Tsoi, A. C. and Back, Andrew D.: Face Detection: A Convolutional Neural-Network Approach, IEEE Transaction n Neural Networks, Vol.8, No.1, (1997) 98-113

6. Hsu, Rein-Lien, Mohamed Abdel-Mottaleb and Jain, Anil K.: Face Detection in Color Images," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.24, No.5, (2002) 696-706
7. Storring, Moritz, Kocka, Tomas, Andersen, Hans J., Granum, Erik: Tracking Regions of Human Skin Through Illumination Changes, Pattern Recognition Letters, Vol.24, (2003) 1715-1723
8. Zhu, Q, Cheng, K. T., Wu, C. T., Wu, Y. L.: Adaptive Learning of an Accurate Skin-Color Model, Proceeding of the Sixth IEEE International Conf. on Automatic Face and Gesture Recognitin, 1 (2004)
9. Kim, Ho-Joon, Yang, Hyun-Seung, "A Modified FMM Neural Network for Pattern Classification and Feature Analysis," Proceeding of the ISATED International Conference on Artificial Intelligence and Applications, 1 (2005)

Fast Learning for Statistical Face Detection

Zhi-Gang Fan and Bao-Liang Lu

Department of Computer Science and Engineering, Shanghai Jiao Tong University,
1954 Hua Shan Road, Shanghai 200030, China
zgfan@sjtu.edu.cn, blu@cs.sjtu.edu.cn

Abstract. In this paper, we propose a novel learning method for face detection using discriminative feature selection. The main deficiency of the boosting algorithm for face detection is its long training time. Through statistical learning theory, our discriminative feature selection method can make the training process for face detection much faster than the boosting algorithm without degrading the generalization performance. Being different from the boosting algorithm which works in an iterative learning way, our method can directly solve the learning problem of face detection. Our method is a novel ensemble learning method for combining multiple weak classifiers. The most discriminative component classifiers are selected for the ensemble. Our experiments show that the proposed discriminative feature selection method is more efficient than the boosting algorithm for face detection.

1 Introduction

Face recognition techniques have been developed over the past few decades. A first step of any face recognition system is detecting the locations in images where faces are present. Face detection has long been an important and active area in vision research. However, face detection from a single image is a challenging task because of variability in scale, location, orientation (up-right, rotated), and pose (frontal, profile). Facial expression, occlusion, and lighting conditions also change the overall appearance of faces. Furthermore, most of the applications of face detection now demand not only accuracy but also real-time response. Viola and Jones proposed an effective coarse-to-fine scheme using boosting algorithm and cascade structure for face detection [17]. Their framework has prompted considerable interest in further investigating the use of boosting algorithm and cascade structure for face detection, e.g., [4], [14], [18], [6], [19], [5], [7].

Sung and Poggio [15] established a face detection approach based on a mixture of Gaussian model. Rowley and Kanade [12] designed a neural network based face detection approach that uses a small set of simple image features. In [9], Osuna *et al.* described an SVM-based method for face detection. Romdhani *et al.* [11] presented another SVM-based face detection system by introducing the concept of reduced set vectors and the sequential evaluation strategy. The SNoW (sparse network of winnows) face detection system by Yang *et al.* [20] is a sparse network of linear functions that utilizes winnows update rules. In

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 187–196, 2006.

[10], Papageorgiou and Poggio established a trainable system for face detection using SVMs and overcomplete Haar wavelet transform. Using an energy-based loss function, Osadchy *et al.* [8] designed convolutional networks for real-time simultaneous face detection and pose estimation. Schneiderman and Kanade [13] established an object detection system using boosting algorithm and wavelet transform.

The excellent work of Viola and Jones [17] has redefined what can be achieved by an efficient implementation of a face detection system. They formulated the detection task as a series of non-face rejection problems. Since then, a number of systems have been proposed to extend the idea of detecting faces through the boosting algorithm. For example, Li *et al.* [4] developed a face detection method through FloatBoost learning. The work by Lienhart and Maydt [5] focused on extending the set of Haar-like features. In [7], Liu and Shum introduced a Kullback-Leibler boosting to derive weak learners by maximizing projected KL distances.

The boosting algorithm is a milestone of the research on face detection. However, the main deficiency of the boosting algorithm for face detection is that a very long training time is required. Using statistical learning theory, we propose a discriminative feature selection method, which can make the training process for face detection much faster than the boosting algorithm without degrading the generalization performance. The boosting algorithm is an iterative learning method, and our discriminative feature selection method can directly solve the learning problem of face detection.

2 Related Work

Viola and Jones [17] have made three key contributions to face detection: Haar-like feature, boosting algorithm and cascade structure. All the three contributions are very important. Haar-like feature is good foundation for image representation in face detection. There are many motivations for using Haar-like features rather than the pixels directly. The most common reason is that Haar-like features can act to encode ad-hoc domain knowledge that is difficult to learn using a finite quantity of training data. Unlike the Haar basis, a set of Haar-like features is overcomplete. So the Haar-like feature can more efficiently represent image in detail than the raw pixel data. Another advantage of using Haar-like feature is that the feature can be rapid calculated using so-called ‘integral image’. The integral image is an intermediate representation for the image which is very similar to the summed area table used in computer graphics for texture mapping. The integral image can be computed from an image using a few operations per pixel. Once computed, any one of these Haar-like features can be computed at any scale or location in constant time.

AdaBoost algorithm was used to select a small number of important features from a huge library of potential Haar-like features [17]. Within any image subwindow the total number of Haar-like features is very large, far larger than the number of pixels. In order to ensure fast classification, the learning process must exclude a large majority of the available features, and focus on a small set of critical features. The goal of feature selection is achieved using AdaBoost learning algorithm by constraining each weak classifier to depend on only a single feature. As a result each stage of the boosting process, which selects a new weak classifier, can be viewed as a feature selection process. The weak learning algorithm is designed to select the single Haar-like feature which best separates the positive and negative examples. For each feature, the weak learner determines the optimal threshold classification function, such that the minimum number of examples are misclassified. A weak classifier $h(x, f, p, \theta)$ thus consists of a feature (f), a threshold (θ) and a polarity (p) indicating the direction of the inequality [17]:

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{if } pf(x) < p\theta \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Here x is a fixed size pixel sub-window of an image.

3 Discriminative Feature Selection

The discriminative feature selection approach proposed in this paper consists of two main steps. The first step is to extract Haar-like features and train single feature weak classifiers, and the second step is to search out a small set of critical features (namely critical weak classifiers) and build classifiers for face detection.

3.1 Feature Extraction

Our feature extraction process uses the Haar-like features as used by Viola and Jones [17]. Being similar to [17], the Haar-like features to be extracted have five prototypes. We also use the weak classifier $h(x, f, p, \theta)$ as shown in equation (1) in our feature extraction process. Unfortunately, as showed in Figure.1, the boosting algorithm for face detection requires all weak classifiers be retrained in each iteration step because the training data have been re-weighted. This is a computationally demanding task which is in the inner loop of the boosting algorithm. Therefore, the boosting algorithm for face detection has very long training time.

As showed in Figure.2, we train all weak classifiers once in advance without retraining the weak classifiers in the afterward discriminative feature selection process. In [19], the same strategy was used and a forward feature selection (FFS) method was proposed for face detection. All weak classifiers $h(x, f, p, \theta)$ are trained on single Haar-like feature after Haar-like feature extraction and the thresholds for every single feature are obtained. By thresholding every single Haar-like feature with these weak classifiers, we set each feature to binary value, zero or one. As a result, the data space becomes a binary value space after feature extraction. Feature selection and classifier construction will be finished within this binary value data space.

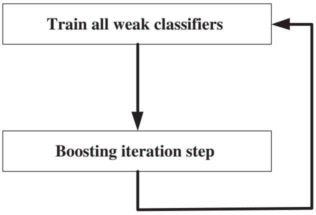


Fig. 1. Weak classifiers training and boosting algorithm

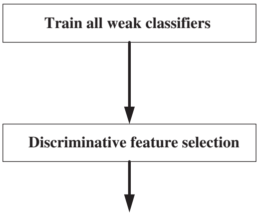


Fig. 2. Weak classifiers training and discriminative feature selection

3.2 Learning and Feature Selection

After feature extraction and thresholding on every single feature by weak classifiers, learning is carried out using statistical learning theory [16] for feature selection and classifier construction in the binary value feature space. So our method is a novel ensemble learning method for combining multiple weak classifiers. Every single feature is a weak classifier in this specific environment. The most discriminative weak classifiers (namely discriminative features) are selected for the ensemble. We use the optimal separating hyperplane in the output space of all the weak classifiers as the combining mechanism for classifier ensemble learning using the statistical learning theory. Statistical learning theory is not only a tool for the theoretical analysis but also a tool for creating practical algorithms for pattern recognition. This abstract theoretical analysis allows us to discover a general model of generalization. On the basis of the VC dimension concept, constructive distribution-independent bounds on the rate of convergence of learning processes can be obtained and the structural risk minimization principle has been found. Optimal separating hyperplane and support vector machines (SVMs) [16] are machine learning techniques which are well-founded in statistical learning theory. As an application of the theoretical breakthrough, SVMs have high generalization ability and are capable of learning in high-dimensional spaces with a small number of training examples. It accomplishes this by minimizing a bound on the empirical error and the complexity of the classifier, at the same time. This controlling of both the training set error and the classifier's complexity has allowed SVMs to be successfully applied to very high dimensional learning tasks.

We are interesting in the optimal separating hyperplane which can also be called linear SVMs because of the nature of the data sets under investigation. Linear SVMs use the optimal hyperplane

$$(w \cdot x) + b = 0 \tag{2}$$

which can separate the training vectors without error and has maximum distance to the closest vectors. In our method, the input vector x is in the output space of all the weak classifiers. We use this optimal separating hyperplane in the output space of all the weak classifiers to combine multiple weak classifiers. To find the optimal hyperplane one has to solve the following quadratic programming problem: minimize the functional

$$\Phi(w) = \frac{1}{2}(w \cdot w) \tag{3}$$

under the inequality constraints

$$y_i[(x_i \cdot w) + b] \geq 1, \quad i = 1, 2, \ldots, l. \tag{4}$$

where $y_i \in \{-1, 1\}$ is class label [16].

According to the hyperplane as shown in equation (2), the linear discriminant function can be constructed for SVMs classifier as follows:

$$f(x) = \operatorname{sign}\{(w \cdot x) + b\} \tag{5}$$

The inner product of weight vector $w = (w_1, w_2, \ldots, w_n)$ and input vector $x = (x_1, x_2, \ldots, x_n)$ determines the value of $f(x)$. Intuitively, the input features in a subset of $(x_1, x_2, \ldots, x_n)$ that are weighted by the largest absolute value subset of $(w_1, w_2, \ldots, w_n)$ influence most the classification decision. If the classifier performs well, the input feature subset with the largest weights should correspond to the most informative features . Therefore, the weights $|w_k|$ of the linear discriminant function can be used as feature ranking coefficients [2], [3], [1]. However, this way for feature ranking is a greedy method and we should look for more evidences for feature selection. In [3] and [1], support vectors have been used as evidence.

Assume the distance between the optimal hyperplane and the support vectors is Δ, the optimal hyperplane can be viewed as a kind of Δ-margin separating hyperplane which is located in the center of margin $(-\Delta, \Delta)$. According to [16], the set of Δ-margin separating hyperplanes has the VC dimension h bounded by the inequality

$$h \leq \min\left(\left[\frac{R^2}{\Delta^2}\right], n\right) + 1 \tag{6}$$

where R is the radius of a sphere which can bound the training vectors $x \in X$ and n is the dimension of the space.

Inequality (6) points out the relationship between margin Δ and VC dimension: a larger Δ means a smaller VC dimension. Therefore, in order to obtain high generalization ability, we should still maintain margin large after feature selection. However, because the dimensionality of original input space has been reduced after feature selection, the margin is usually to shrink and what we can do is trying our best to make the shrink small to some extent. Therefore, in feature selection process, we should preferentially select the features which make more contribution to maintaining the margin large. This is another evidence for feature ranking. To realize this idea, a coefficient is given by

$$c_k = \left| \frac{1}{l_+} \sum_{i \in SV_+} x_{i,k} - \frac{1}{l_-} \sum_{j \in SV_-} x_{j,k} \right| \tag{7}$$

where SV_+ denotes the support vectors belong to positive samples, SV_- denotes the support vectors belong to negative samples, l_+ denotes the number of SV_+, l_- denotes the number of SV_-, and $x_{i,k}$ denotes the kth feature of support vector i in input space R^n.

The larger c_k indicates that the kth feature of feature space can make more contribution to maintaining the margin large. Therefore, c_k can assist $|w_k|$ for feature ranking. The solution is that, combining the two evidences, we can order the features by ranking $c_k|w_k|$ and select the features which have larger value of $c_k|w_k|$. We present below an outline of the discriminative feature selection and classifier training algorithm.

```
• Input:
  Training examples (using binary Haar-like features)
      $X_0 = \{x_1, x_2, \dots x_l\}^T$
• Initialize:
  Indices for selected features:   $s = [1, 2, \dots n]$
  Train the SVM classifier using samples $X_0$
• For $t = 1, \dots, T$ :
    1. Compute the ranking criteria $c_k|w_k|$ according to the trained SVMs
    2. Order the features by decreasing $c_k|w_k|$, select the top $M_t$ features, and
       eliminate the other features
    3. Update $s$ by eliminating the indices which not belong to the selected
       features
    4. Restrict training examples to selected feature indices
          $X = X_0(:, s)$
    5. Train the SVM classifier using samples $X$
• Outputs:
  The small set of critical features and the final SVM classifier
```

Usually, the iterative loop in the algorithm can be terminated before the training samples can not be separated by a hyperplane. Clearly, this algorithm can integrate the two tasks, feature selection and classifier training, into a single consistent framework and make the feature selection process more effective. Using this discriminative feature selection method, we can search out the small set of critical features and build classifiers for face detection.

4 Experiments

We have made several sets of experiments to illustrate the effectiveness of the proposed discriminative feature selection algorithm for face detection. In all experiments reported here, we use the MIT-CBCL face database [3] , a database of faces and non-faces that have been used extensively at the Center for Biological and Computational Learning at MIT. All input gray-scale images are of size

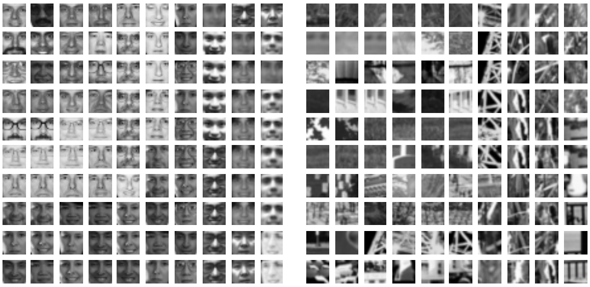


Fig. 3. Some face and non-face sample images in the MIT-CBCL database

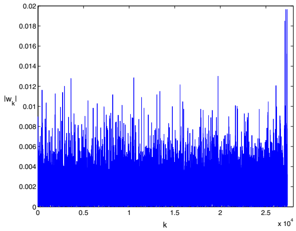


Fig. 4. The diversity of $|w_k|$

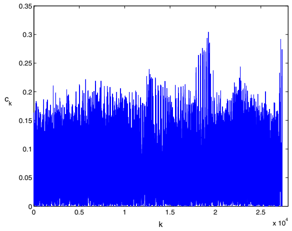


Fig. 5. The diversity of c_k

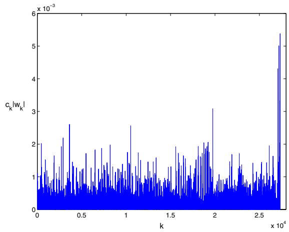


Fig. 6. The diversity of $c_k|w_k|$

19×19 and the dimensionality of the resulting input vectors is $N = 361$. Figure 3 depicts some face and non-face sample images in the MIT-CBCL database. The overall database is partitioned into two subsets: the training set and test set. The training set is composed of 2429 face images and 4548 non-face images. The test set is composed of 472 face images and 23573 non-face images. All the image data have been histogram equalized . All of the experiments were performed on a 3.0GHz Pentium 4 PC with 2.0GB RAM.

After Haar-like feature extraction, the dimensionality of the feature vectors without feature selection is $N = 27348$. In the binary value feature space of the dimensionality $N = 27348$, we train linear SVMs and obtain the coefficients c_k and $|w_k|$. The diversities of c_k, $|w_k|$ and $c_k|w_k|$ have been showed in Figures.4 through 6, respectively. Figures 7 through 9 show, respectively, c_k, $|w_k|$ and $c_k|w_k|$ being ordered increasingly. From these figures, we can see that $c_k|w_k|$ has the steepest variability curve which is useful for feature selection. To evaluate the different impacts of the three coefficients on feature selection, we use the three coefficients respectively to select features. We use four iterative steps (T= 4) and the parameter M_t is set as: $M_1 = 5000, M_2 = 1000, M_3 = 500, M_4 = 200$. After feature selection, the classification accuracy is examined on the test data set. The test results are showed in Table 1, where 'FS-W', 'FS-C', and 'FS-CW' denote the feature selection using coefficient $|w_k|$, the feature selection using coefficient c_k, and the feature selection using coefficient $c_k|w_k|$, respectively. The

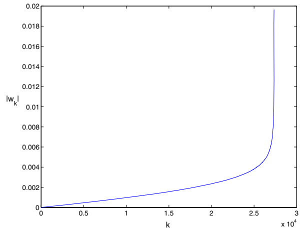


Fig. 7. $|w_k|$ ordered increasingly

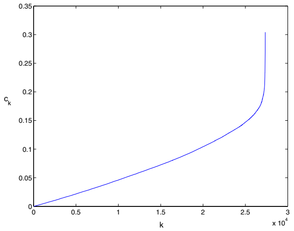


Fig. 8. c_k ordered increasingly

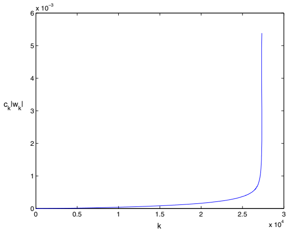


Fig. 9. $c_k|w_k|$ ordered increasingly

Table 1. Test results using three coefficients respectively

Methods	No. features	True positive rate (%)	True negative rate (%)
FS-W	5000	42.3729	98.8080
	1000	41.3136	98.7231
	500	41.1017	98.7274
	200	41.1017	98.4643
FS-C	5000	42.1610	98.5110
	1000	41.1017	94.8119
	500	40.4661	93.3271
	200	39.6186	95.3167
FS-CW	5000	42.5847	98.9098
	1000	41.9492	98.7443
	500	41.5254	98.8589
	200	41.5254	98.5025

linear SVMs are used as classifier in the three cases. Through Table 1 we can see that the FS-CW approach is the best one among the three methods.

Figure 10 shows the ROC (receiver operating characteristic) curves for the face detection test. In this set of experiments, we have used four different methods for comparison study. In Figure 10, ‘FFS’, ‘Viola-Boosting’, and ‘Pixel Method’ denote the forward feature selection method [19], the AdaBoost algorithm [17], and the linear SVMs using raw pixel data, respectively. The experimental setting of our method is the same as mentioned above. We used linear SVMs as the weight setting algorithm of the FFS method. In the pixel method, we used the raw image pixel data as input features and didn’t use the Haar-like features. But the Haar-like features have been used for the FFS and Viola-Boosting. The dimensionality of the raw pixel feature vectors is $N = 361$ and the parameter C of the linear SVMs was set to 0.001 for the pixel method. For the other three methods, our discriminative feature selection method, FFS and the Viola-Boosting, the dimensionality of the feature vectors is $N = 200$ after feature

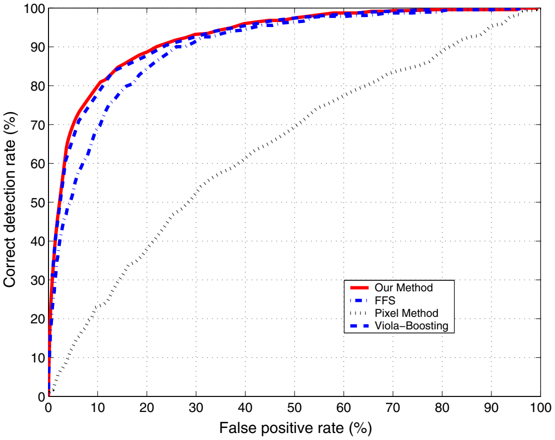


Fig. 10. ROC (receiver operating characteristic) curves for the face detection test

selection. Through Figure 10, we can see that the accuracy of our method is the highest among the four methods. And our method has much shorter training time than the Viola-Boosting algorithm. In our experiments, the training time of our discriminative feature selection method is 15 minutes and the training time of Viola-Boosting is 7 hours. The accuracy of the pixel method is very low because it doesn't use Haar-like features.

5 Conclusions

We have presented a discriminative feature selection method for face detection. This discriminative feature selection method can make the training process for face detection much faster than the boosting algorithm without degrading the generalization performance. The boosting algorithm works in an iterative way, while our discriminative feature selection method can directly solve the learning problem of face detection. Our method is a novel ensemble learning method for combining multiple weak classifiers. We use the optimal separating hyperplane in the output space of all the weak classifiers as the combining mechanism for classifier ensemble learning. The most discriminative component classifiers are selected for the ensemble. Through the experimental results, we can see that our method is more efficient than the boosting algorithm for face detection. We also can see that the Haar-like features are more powerful than the raw pixel features. We can learn more detail of the nature of the learning methods for face detection in this study.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China via the grants NSFC 60375022 and NSFC 60473040.

References

1. Z. G. Fan and B. L. Lu. Fast recognition of multi-view faces with feature selection. *Proc. ICCV 2005*, 1:76–81, 2005.
2. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(3):389–422, 2002.
3. B. Heisele, T. Serre, S. Prentice, and T. Poggio. Hierarchical classification and feature reduction for fast face detection with support vector machine. *Pattern Recognition*, 36(9):2007–2017, 2003.
4. S. Z. Li and Z. Zhang. Floatboost learning and statistical face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1112–1123, 2004.
5. R. Lienhart and J. Maydt. An extended set of haar-like features for papid object detection. *Proc. ICIP 2002*, 1:900–903, 2002.
6. Y. Lin and T. Liu. Robust face detection with multi-class boosting. *Proc. CVPR 2005*, 1:680–687, 2005.
7. C. Liu and H. Shum. Kullback-leibler boosting. *Proc. CVPR 2003*, 1:587–594, 2003.
8. R. Osadchy, M. Miller, and Y. LeCun. Synergistic face detection and pose estimation with energy-based model. In *Advances in Neural Information Processing Systems (NIPS 2004)*. MIT Press, 2005.
9. E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. *Proc. CVPR 1997*, 1:130–136, 1997.
10. C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.
11. S. Romdhani, P. Torr, B. Scholkopf, and A. Blake. Computationally efficient face detection. *Proc. ICCV 2001*, 2:695–700, 2001.
12. H. Rowley and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.
13. H. Schneiderman and T. Kanade. Object detection using the statistics of patrs. *International Journal of Computer Vision*, 56(3):151–177, 2004.
14. J. Sun, J. M. Rehg, and A. Bobick. Automatic cascade training with perturbation bias. *Proc. CVPR 2004*, 2:276–283, 2004.
15. K. K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.
16. V. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag New York, 2000.
17. P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.
18. B. Wu, H. Ai, C. Huang, and S. Lao. Fast rotation invariant multi-view face detection based on real adaboost. *Proc. FGR 2004*, 1:79–84, 2004.
19. J. Wu, J. M. Rehg, and M. D. Mullin. Learning a rare event detection cascade by direct feature selection. In *Advances in Neural Information Processing Systems 16.* MIT Press, Cambridge, MA, 2004.
20. M. H. Yang, D. Roth, and N. Ahuja. A snow-based face detector. In *Advances in Neural Information Processing Systems 12.*

Extraction of Discriminative Manifold for Face Recognition

Yanmin Niu[1] and Xuchu Wang[2]

[1] College of Physics and Information Techniques, Chongqing Normal University,
Chongqing 400047, China
niuym@cqnu.edu.cn

[2] Key Lab on Opto-Electronic Technique and Systems, Ministry of Education,
Chongqing University, Chongqing 400044, China
seadrift.wang@gmail.com

Abstract. It is very meaningful for dimension reduction by extraction and analysis of the underlying manifold embedded in face observation space, since the low dimensional manifold can represent the varying intrinsic features. However, this kind of manifold is perhaps not useful for face image recognition problem. This paper proposes a new discriminative manifold learning method which can efficiently discover the discriminative manifold. Besides the characteristic of preserving the local structure similarity in the face submanifold, the proposed method emphasizes the discriminative property of embedding much more throughout building and solving an object function. Experimental results on some open face datasets indicate the proposed method can achieve lower error rates.

1 Introduction

It is a challengeable task to reduce the influence of pose, illumination and expression in the field of face representation and recognition. Although Principal Components Analysis (PCA)-based [1] and Linear Discriminant Analysis (LDA) -based [2] methods have been effectively applied to extract facial features, they fail to represent this kind of nonlinear structure and are hard to get a higher performance. There are two methods to solve this problem, one is kernel-based method, the basic idea of this method is to map the points in observation space into a higher dimensional feature space by a kernel function, where the points are assumed linearly separable [3]. Due to the success of kernel function in Support Vector Machine (SVM), the nonlinear feature extracted by this method is helpful for face recognition, and there appears many nonlinear face recognition methods such as Kernel PCA (KPCA) [4], Kernel LDA (KDA) [5] and improved versions. However, most of these methods are not only computationally expensive, too implicit for choosing parameters, but also only extract the global nonlinear structure.

The other is based on manifold learning. The fields of face recognition and computer vision have witnessed recently growing interests in manifold learning. From this viewpoint, faces are thought of data points possibly residing close to a nonlinear submanifold embedded in a high-dimensional observation space.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 197–206, 2006.

Some nonlinear techniques i.e. Isomap [6], LLE [7] and Laplacian Eigenmaps [8], have been proposed to discover the nonlinear structure of the manifold. These nonlinear methods do yield impressive results on some benchmark artificial datasets. However, they are developed based on reconstruction and perhaps are not optimal for classification purpose [9][10][11]. Moreover, they are difficult for new-come data which is essential for face or digital number recognition. In order to cope with this problem, Yang proposed an extended Isomap method[9] that utilized LDA to replace Multidimensional Scaling (MDS) during the low-dimensional embedding process. He and Niyogi proposed a Local Preserving Projections (LPP) method [10][11], which is an optimal linear approximation to Laplacian Beltrami operator on the face manifold, and very flexible in connection with both PCA/LDA versus clustering/classification. However, LPP shares local preserving character to LLE, which still goes against in face recognition in some sense. Chen and Chang proposed Local Discriminant Embedding (LDE) [12] and extension versions which seek to dissociate the submanifold of each class from one another, and outperform than many classical methods.

The proposed method in this paper focuses on the relationships of some subspace analysis and manifold learning from the viewpoint of supervised classification, moreover, it owns more discriminative ability essentially to different face classes. In order to discover the discriminative manifold, an natural extension of Fisher LDA criterion in manifold sense, called Fisher Manifold Discriminant Embedding (MDE), is introduced to build an object function. The experimental results on three open datasets show the effectiveness and superiority of our method designed by this criterion.

The remainder of the paper is structured as follows. Section 2 reviews the some subspace-based methods and manifold learning methods for face recognition. Section 3 comments the new algorithm through Fisher Manifold Discriminative Embedding criterion analysis. Section 4 describes the experimental results based on some open datasets and discussions. Section 5 summarizes the paper and indicate the main interests for future work.

2 Simple Review of Subspace-Based Face Recognition

Since last decade, subspace-based methods, originated from Turk's Eigenfaces [1] based on the PCA and improved by Belhumeur's Fisherfaces [2] based on Fisher LDA, have dominated the approaches in face recognition for good performance and computational feasibility. They both try to transform a given set of face images into a smaller set of basis images using matrix decomposition techniques. While the unsupervised Eigenface intends to maximize the covariance and the supervised Fisherface intends to maximize the discriminability.

Suppose $\{\omega_i\}_{i=1}^{c}$ are c known pattern classes, $\{x_i\}_{i=1}^{N}$ are N h-dimensional samples, n_i is the number of samples in the subset ω_i. Let m be the mean sample of all samples and be the mean for the i-th class, then we can calculate the between-class scatter matrix S_b, the within-class scatter matrix S_w and the total scatter matrix S_t. PCA finds orthogonal transform with the basis $\Phi = \{\phi_i\}_{i=1}^{K}$

that for any $K \ll N$ minimizes the reconstruction error, in other words, The objective function is

$$\arg\max J_K(\Phi) = |\Phi^T S_t \Phi| \tag{1}$$

Whereas as a linear statistic classification method, Fisher LDA tries to find a linear transform W so that after its application the scatter of sample vectors is minimized within each class and the scatter of mean vectors around the total mean vector is maximized simultaneously. It can be formulated as an optimization problem of S_b and S_w, and the objective function is highlighted as follows:

$$\arg\max J_F(W) = \frac{|W^T S_b W|}{|W^T S_w W|} \tag{2}$$

Although Martinez explained that LDA doesn't always outperform than PCA [13], LDA is still widely accepted in face recognition and more effective than PCA. In many practical applications, there are not enough samples to make the within-class scatter matrix nonsingular (i.e. small sample size problem, SSSP) and S_w is ill-posed. In order to cope with this problem, Belhumeur [2] use PCA to reduce dimensionality. Yang [14] proposed a direct LDA method to diagonalize the S_b and S_w. Chen [15] regarded the null space of S_w was particularly useful in discriminability and proposed a way to makes use of it. Huang [16] followed this basic idea by firstly removing the common null space of both S_b and S_w, which means the null space of total-scatter matrix S_t is also removed since $S_t = S_b + S_w$.

We can find from the above analysis, the similarity and dissimilarity property versus same class and different class should be equally considered, and their characteristic of singularity is useful for classification purpose. This idea is also can be introduced to manifold learning-based face recognition.

3 Discriminative Manifold for Face Recogntion

From viewpoint of manifold learning, $\mathcal{M}$ is supposed as a manifold embedded in $\mathbb{R}^h$ and $\{x_i \in \mathbb{R}^h\}_{i=1}^N$, any subset of data points that belong to the same class is assumed to lie in a submanifold of $\mathcal{M}$. Local Preserving Projection defines an objective function

$$\arg\min J_S(\mathbf{w}) = \sum\nolimits_{i,j} (\mathbf{w}^T(x_i - x_j))^2 s_{ij} \tag{3}$$

to discover the preserving submanifold. By changing the similarity matrix S, it can correspond to the method of PCA and LDA. However, it cannot share their properties simultaneously. Here we want to discover the most of a discriminative submanifold for classification, dimensionality reduction and etc, so the object becomes the discovery of a most preserving or discriminative submanifold and the following fact should be respected: if two data points are close, we hope them still keep close in submanifold and vice versa. Of course, when class information is labeled, there are some points are close in different class and are far away in same class because of the noisy point and the outliers. According to the basic manifold assumptions, similarities can be locally measured, and in order to emphasis the similarity and dissimilarity among the neighborhoods of a point, we use two neighborhood graphs to measure this locality under the constraint of class information. They are defined as similar in [12]:

Let $Nb = \{x_j\}_{j=1}^{b}$ is a subset of b nearest neighbors of a data point x_i , G and $\overline{G}$ denote two undirected graphs both over all points. We consider each pair of points x_i, x_c and $x_c \in Nb$, when they are from same class an edge is added to between x_i, x_c (The ε -ball implementation way also can be considered). When they are from different classes, an edge is added to $\overline{G}$ between x_i, x_c.

According to neighborhood graphs G and $\overline{G}$, the affinity matrix S and $\overline{S}$ can be specified, where each element s_{ij} refers to the weight of the edge between x_i, x_j in S ,and refers $\overline{s}_{ij}$ to $\overline{S}$. The weight can be given by the way of "heat kernel", "cosine kernel" or "simple-minded". For example, a cosine kernel is a similarity distance measure as follows:

$$s_{ij} = \begin{cases} \frac{<x_i,x_j>^2}{<x_i,x_i><x_j,x_j>} & \text{if } x_i, x_j \text{ are connected in G} \\ 0 & \text{else} \end{cases} \tag{4}$$

where $< \cdot, \cdot >$ means a kind of inner product operation. An advantage of cosine kernel is it doesn't need to adjust parameter. Just like the Fisher LDA in Eq.2, we propose a Fisher Manifold Discriminant Embedding (MDE) criterion $\overline{J}_M(W)$ for classification purpose:

$$\arg\max \overline{J}_M(W) = \frac{\sum_{i,j} \|y_i - y_j\|^2 \overline{s}_{ij}}{\sum_{i,j} \|y_i - y_j\|^2 s_{ij}} \tag{5}$$

where y_i is the shape of x_i after manifold embedding, $y_i = W^T x_i$, and $\|y_i - y_j\|^2$ is the difference measure of y_i and y_j in a difference matrix. It wants to find a transform which can minimize the within-class difference and maximize the between-class difference of the face submanifolds simultaneously. The optimization problem can be solved as follows. Let $J_M(W) = \sum_{i,j} \|W^T x_i - W^T x_j\|^2 s_{ij}$, so

$$\begin{aligned} J_M(W) &= \sum\nolimits_{i,j} (W^T x_i - W^T x_j)^T (W^T x_i - W^T x_j) s_{ij} \\ &= 2 \sum\nolimits_{i,j} W^T (x_i s_{ij} x_i^T - x_i s_{ij} x_j^T) W \\ &= 2(W^T X D X^T W - W^T X S X^T W) \\ &= 2 W^T X (D - S) X^T W \end{aligned} \tag{6}$$

Let Laplacian matrix $L = D - S$, $\overline{L} = \overline{D} - \overline{S}$ and the fact that L is symmetric and positive semidefinite causes XLX^T is symmetric and positive semidefinite. So $J_W(W) \geq 0$, and the Fisher MDE criterion can be analyzed from the following two aspects:

(1) $J_W(M) > 0$. This case happens when L is non-singular. The solution is similar as Local Discriminant Embedding (LDE) and we omit this procedure, the

embedding matrix $W = [w_1, w_2, ..., w_k]$ can be obtained by solving the following generalized eigenvector problem:

$$X\overline{L}X^T w = \lambda XLX^T w \tag{7}$$

where $\overline{J}_M(W)$ is a max finite real number in $\mathbb{R}$.

(2) $J_W(M) = 0$. In this case, consider the property of symmetric and positive semidefinite matrix:

$$\begin{aligned} J_M(W) = 2W^T X(D - S)X^T W = 0 \\ \Leftrightarrow X(D - S)X^T W = 0 \\ \Leftrightarrow X(D - S)X^T = 0 \\ \Leftrightarrow (D - S)X^T = 0 \end{aligned} \tag{8}$$

It means that mapping the data points to the null space of L can make $J_M(W) = 0$. $L \in \mathbb{R}^{n\times n}$, the rank of L is $n-c$ [11]. so the dimensionality of the null space of L is $c-1$. Let $W_{null} = \{v_i\}_{i=1}^{c-1}$ is the basis set which spans the null space of L, so firstly after consideration of using PCA as preprocessing for noise reduction(note that the reconstruction is without any loss. For simplicity, we still use X to represent the original data points after PCA dimensionality reduction). We then project the data $X \rightarrow \tilde{X}$,$\tilde{X} = W_{null}^T X$, and change the Fisher MDE criterion as follows:

$$\arg\max \tilde{\overline{J}}_M(W) = \sum\nolimits_{i,j} \|\tilde{y}_i - \tilde{y}_j\|^2 \overline{s}_{ij} \tag{9}$$

where $\tilde{y}_i = W^T \tilde{x}_i$, and apparently, $\tilde{\overline{J}}_M(W) = W^T W_{null}^T X\overline{L}X^T W_{null} W$. Since $W_{null}^T X\overline{L}X^T W_{null}$ is a full rank matrix, we can solve the optimization problem by finding the generalized eigenvectors $\{w_i\}_{i=1}^k$ corresponding to the k largest eigenvalues of $W_{null}^T X\overline{L}X^T W_{null}$ and $W = \{w_i\}_{i=1}^k$. So the embedding matrix is $W_{null}W$ after PCA process. Theoretically speaking, it is notable that $\overline{J}_M(W) \rightarrow +\infty$ and the discriminability is the best one.

4 Experimental Results and Discussion

Here we compared our proposed method with the several other face recognition methods (Eigenface, Fisherface, Null-space LDA, LPP and LDE) using the publicly available Yale, AT&T and CMU PIE database. Our intension is to discover different characteristics among these methods.

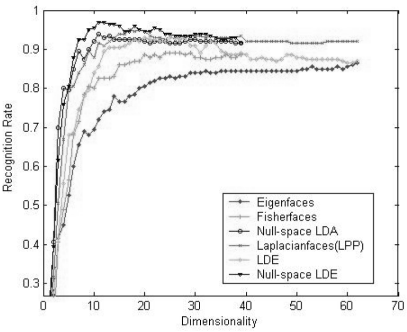
The AT&T database [17] contains 400 images of 40 persons where the variations are mainly due to the facial contours, scale and pose of a person in the image. The Yale database [18] contains 165 images of 15 individuals where the images demonstrate variations in lighting condition, face expression, and with/without glasses. The CMU PIE database [19] contains 41368 face images of 68 subjects under varying pose, illumination and expression. Some processes during the experiment are marked as follows:

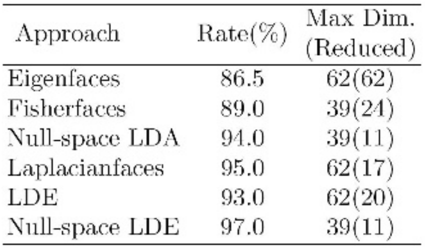
(I)For each image in Yale database, we manually crop the face to size of 92 × 112 (same as the resolution in AT&T). For computational efficiency, each image in both two databases is down-sampled to 1/4 of the original size. For PIE database, we use the dataset collected by He [11]. Which means each face image is cropped to 32×32 sizes and one individual holds 170 images. We lastly normalize them to be zero-mean and unit-variance vectors.

(II)The parameters, such as the number of principle components for dimensionality reduction in Eigenface, LPP and LDE methods, are empirically determined to achieve the lowest error rate by each method. So at last, the dimensionality of projection is different among these methods. The neighbor number for each test is same as the training number of per subject. For Fisherface and our proposed NLDE method, the projection dimensionality are both $c - 1$. The recognition is performed using nearest-neighbor (1-NN) classifier for its simplicity. And the number for training/test is changeable for different purpose.

The experimental details are discussed as follows:

(1) Experiment on the AT&T Database: We firstly use a case to compare the performance of different methods where first 5 images of each individual for training and the rest for testing. For those methods need PCA to reduce dimensionality firstly, the number of principal components is decided by remaining 95% energy. The heat kernel with same parameter is designed to measure the affinity matrix. The recognition results are shown in Fig.1.





Approach	Rate(%)	Max Dim. (Reduced)
Eigenfaces	86.5	62(62)
Fisherfaces	89.0	39(24)
Null-space LDA	94.0	39(11)
Laplacianfaces	95.0	62(17)
LDE	93.0	62(20)
Null-space LDE	97.0	39(11)

Fig. 1. A case of recognition accuracy versus dimensionality on AT&T database (first 5 images of each individual for training and the other for testing).The right is best recognition rate corresponding to dimension reduction.

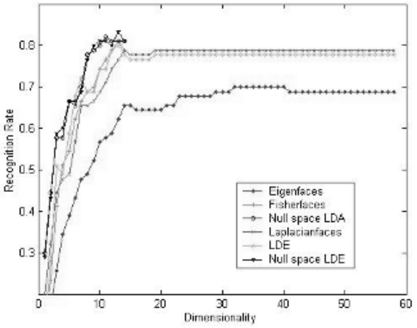
From Fig.1 we can find that the performance of Null-space LDA is better than Eigenfaces and Fisherfaces because it considers the most discriminative vector in the null space of the within-class scatter matrix S_w. Among the manifold-based methods, Null-space LDE method outperforms the others. We also find that the recognition curve of Eigenfaces and Laplacianfaces is similar;and that of Null-space LDA and Null-space LDE is similar, too.

We further repeat 10 times to get the average values of the best recognition rate of each method under different training samples. The result is reported in Table 1. Here we preserve the number of principal components for 98% energy.

Table 1. Performance comparison on AT&T database, each method is gotten from the average best recognition rate of 10 times under different training samples by random selection

Train. num. Approach	2	3	5	7	9
Eigenfaces(PCA)	72.19(60)	78.75(75)	84.89(108)	82.62(80)	82.57(33)
Fisherfaces(PCA+LDA)	76.25(25)	83.21(39)	93.44(33)	95.83(32)	90.00(17)
Null-space LDA (LDA)	82.18(39)	88.93(39)	96.00(37)	97.46(38)	97.50(36)
Laplacianfaces (PCA+LPP)	81.16(48)	87.50(36)	96.02(31)	95.83(37)	99.78(12)
LDE (PCA+LDE)	75.94(63)	84.29(43)	93.52(51)	95.92(24)	99.78(13)
Null-space LDE	83.13(35)	90.97(39)	97.01(33)	97.68(37)	99.84(13)

(2) Experiment on the Yale database: The subjects in this database is much less than AT&T dataset, while the illumination condition is more complex. For those methods use PCA to reduce dimensionality, we preserve 95% energy, Fig.2 shows a case of recognition curves of different methods, where we still can find the the distinctive characters mentioned above. Table 2 reports the average best performance of different methods by 10 times experiments.



Approach	Rate(%)	Max Dim. (Reduced)
Eigenfaces	70.0	58(32)
Fisherfaces	81.11	14(13)
Null-space LDA	82.22	14(11)
Laplacianfaces	78.89	58(14)
LDE	80.0	58(13)
Null-space LDE	83.33	14(13)

Fig. 2. A case of recognition accuracy versus dimensionality on Yale database (first 5 images of each individual for training and the other for testing). The right is best recognition rate corresponding to dimension reduction.

(3) Experiment on the CMU PIE database: This selected dataset only contains five near frontal poses (C05, C07, C09, C27, C29) and all the images under different illuminations and expressions. So, there are 170 images for each individual. In the stage of case study, we use five images of each subject for training and the other five images of each subject for testing. Fig.3 depicts twenty images of two individuals, for each subject, the upper row is for training and the down row is for testing.

For those methods need PCA to dimensionality reduction firstly, we decide the number of principal components by remaining the 95% energy. The affine matrix is based on heat kernel. Fig.4 shows the recognition curves of various methods

Table 2. Performance comparison on Yale database, each method is gotten from the average best recognition rate of 10 times under different training samples by random selection

Train. num. Approach	2	3	5	7	9
Eigenfaces(PCA)	54.07(22)	55.56(29)	59.17(59)	76.67(27)	93.33(44)
Fisherfaces(PCA+LDA)	48.89(11)	76.67(13)	78.89(10)	86.67(14)	90.00(14))
Null-space LDA (LDA)	57.04(14)	78.33(14)	81.10(9)	88.33(11)	93.33(9)
Laplacianfaces (PCA+LPP)	61.00(12)	79.17(15)	77.78(10)	88.33 (14)	98.88(14)
LDE (PCA+LDE)	758.52(26)	74.17(18)	78.89(10)	88.33(13)	98.93(13)
Null-space LDE	61.96(14)	78.50(14)	82.22(13)	90.00 (9)	98.96(10)

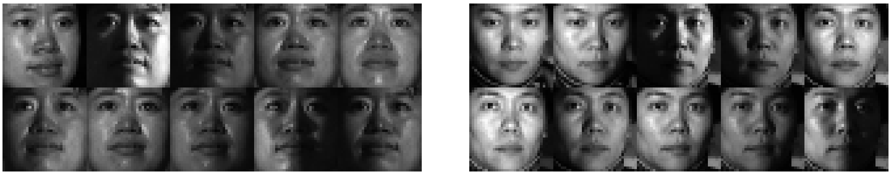
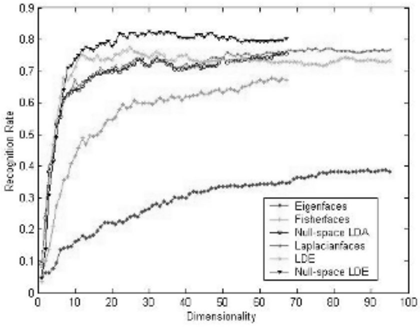
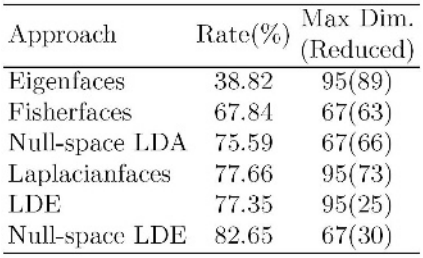


Fig. 3. The cropped face image samples of two subjects on PIE database

and Table 3 reports the average best performance of different methods by 10 times experiments. For those methods need PCA to dimensionality reduction firstly, we decide the number of principal components by preserving the 95% energy. Considering the computational cost, we choose the cosine kernel-based affine matrix because it doesn't need adjust parameters. Although the training number of each individual can be improved to 169, we only select three kinds (5, 10 and 15) for experiment, and which is equal to the testing number per class.

(4) Discussion: Three experiments on three datasets have been carried out. On each dataset, we firstly use a case to compare different methods and prepare for parameter selection. Although it is not sure that we have chosen the best parameter for each evaluation and the results perhaps are not the best, we still try to keep the impartiality of each method, and some discussions are drawn as follows: (I)Eigenfaces and Laplacianfaces both use PCA to reduce dimensionality and preserve the very vectors for reconstruction in a matrix which describes the assimilability in each class. PCA uses S_t, while Laplacianfaces uses S_t and L. The optimization objects of Null-space LDA and Null-space LDE are similar; they get a set of most discriminative vectors in nulls space of the matrices S_w and L which describes the dissimilarity each class, furthermore, Null-space LDA add the dissimilarity to the $\overline{L}$. So it can get the promising performance among all the methods. (II)The performance of methods considering face submanifold is better than those not considering this local structure. However, when the training number of each individual is not very large, null-space methods, even not considering face submanifold, give more effective results and vice versa. So combining the advantage of null-space and discriminative manifold can yield impressive results. (III)From the viewpoint of dimensionality reduction, all the methods based on





Approach	Rate(%)	Max Dim. (Reduced)
Eigenfaces	38.82	95(89)
Fisherfaces	67.84	67(63)
Null-space LDA	75.59	67(66)
Laplacianfaces	77.66	95(73)
LDE	77.35	95(25)
Null-space LDE	82.65	67(30)

Fig. 4. A Case of recognition accuracy versus dimensionality on PIE database (the 5 images of each individual for training are 1,4,7,10,13 and the images for testing are 2, 5, 8, 11, 14). The right is best recognition rates corresponding to dimension reduction.

Table 3. Performance comparison on PIE database, each method is gotten from the average best recognition rate of 10 times under different training samples by random selection

Train. num. / Approach	5	10	15
Eigenfaces(PCA)	39.17(90)	56.33(118)	61.14(123)
Fisherfaces(PCA+LDA)	67.89(61)	77.28(62)	83.46(65)
Null-space LDA (LDA)	75.21(64)	78.68(65)	84.20(58)
Laplacianfaces (PCA+LPP)	78.30(75)	79.31(89)	86.92(115)
LDE (PCA+LDE)	78.83(26)	80.26(32)	87.27(60)
Null-space LDE	81.07(28)	82.35(33)	87.43(61)

manifold learning can achieve the presetting object more quickly. While some methods such as PCA are not so effective. Moreover, PCA-based method is always regarded without consideration of discriminability, however, in some case, it actually outperforms than LDA-based methods as discussed by Martinez.

5 Conclusion and Future Work

In this paper, a discriminative manifold learning method for face recognition is introduced. The basic idea of this method can be modeled by a Fisher Manifold Discriminant Embedding (MDE) criterion. Its implementation is similar with LPP, LDE, but simpler, faster and more powerful in face recognition. From the combined viewpoint of discriminant analysis and manifold learning, the proposed method can discover the most discriminative nonlinear structure of the face images and is an optimal solution for face recognition. Of course, it can be extended by kernel methods, 2D representation and etc. Which is our next main research interests.

References

1. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience **3**(1) (1991) 71–86
2. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7) (1997) 711–720
3. Li, S.Z., Jain, A.K., eds.: Handbook of Face Recognition. Springer-Verlag (2004)
4. Schölkopf, B., Smola, A.J., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation **10**(5) (1998) 1299–1319
5. Mika, S., Ratsch, G., J.Weston: Fisher discriminant analysis with kernels. In: Proc. IEEE Neural Networks for Signal Processing, USA (1999) 41–48
6. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensional reduction. Science **290**(5500) (2000) 2319–2323
7. Roweis, S., L.K.Saul: Nonlinear dimensional reduction by locally linear embedding. Science **290**(5500) (2000) 2323–2326
8. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In Dietterich, T.G., Becker, S., Ghahramani, Z., eds.: Advances in Neural Information Processing Systems(NIPS), MIT Press (2001) 585–591
9. Yang, M.H.: Face recognition using extended isomap. In: ICIP (2). (2002) 117–120
10. He, X., Niyogi, P.: Locality preserving projections. In Thrun, S., Saul, L.K., Schölkopf, B., eds.: Advances in Neural Information Processing Systems(NIPS), MIT Press (2003)
11. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.J.: Face recognition using Laplacianfaces. IEEE Trans. Pattern Anal. Mach. Intell. **27**(3) (2005) 328–340
12. Chen, H.T., Chang, H.W., Liu, T.L.: Local discriminant embedding and its variants. In: CVPR (2), IEEE Computer Society (2005) 846–853
13. Martinez, A.M., Kak, A.C.: PCA versus LDA. IEEE Trans. Pattern Anal. Mach. Intell. **23**(2) (2001) 228–233
14. Yang, J., Yu, H., Kunz, W.: An efficient LDA algorithm for face recognition. In: 6th International Conference on Control, Automation, Robotics and Vision (ICARCV2000), Singapore (2000)
15. Chen, L.F., Liao, H.Y.M., Ko, M.T., Lin, J.C., Yu, G.J.: A new LDA-based face recognition system which can solve the small sample size problem. Pattern Recognition **33**(10) (2000) 1713–1726
16. Huang, R., Liu, Q., Lu, H., Ma, S.: Solving the small sample size problem of LDA. In: ICPR (3). (2002) 29–32
17. AT&T Lab: AT&T face database (2002) http://www.uk.research.att.com/facedatabase/.
18. Yale Univ.: Face database (2002) http://cvc.yale.edu/projects/yalefaces/.
19. Sim, T., Baker, S., Bsat, M.: The CMU database. IEEE Trans. Pattern Anal. Mach. Intell. **25** (2003) 1615–1618

Gender Classification
Using a New Pyramidal Neural Network

S.L. Phung and A. Bouzerdoum

University of Wollongong, Northfields Av, Wollongong, NSW 2522, Australia
{phung, salim}@uow.edu.au

Abstract. We propose a novel neural network for classification of visual patterns. The new network, called *pyramidal neural network* or PyraNet, has a hierarchical structure with two types of processing layers, namely pyramidal layers and 1-D layers. The PyraNet is motivated by two concepts: the image pyramids and local receptive fields. In the new network, nonlinear 2-D are trained to perform both 2-D analysis and data reduction. In this paper, we present a fast training method for the PyraNet that is based on resilient back-propagation and weight decay, and apply the new network to classify gender from facial images.

1 Introduction

Artificial neural networks (ANNs) have found applications in many tasks: pattern classification, function approximation, data clustering, and data compression, to name a few. Apart from parallel processing and noise tolerance capabilities, ANNs are able to learn from examples in a similar way as their biological counterparts. Hence, they are suitable for problems in which the solutions are either impossible or difficult to express analytically. In machine vision, neural networks have been used for numerous visual recognition tasks, eg. hand-written digit recognition [1] and facial image analysis [2].

Among the neural networks proposed for visual recognition, two significant models are the neocognitron and convolutional neural networks. The neocognitron, introduced by Fukushima [3], is a hierarchical neural network motivated by a model of the visual cortex in mammals, which was proposed by Hubel and Wiesel. It consists of two main types of cells: the S-cells model the feature extraction in the simple cortical cells whereas the C-cells model the information processing performed by complex cortical cells. Convolutional neural networks, developed by LeCun and his colleagues [1], are built upon three key ideas, namely receptive fields, weight sharing and subsampling in spatial/temporal domain. Both the neocognitron and convolutional neural networks retain the 2-D representation of images in analysis stages.

In this paper we propose a new neural network model for visual recognition, called *pyramidal neural network* or PyraNet for short. The new neural network is motivated by the image pyramids that have been used successfully for image processing tasks such as image decomposition, image segmentation, and image compression [4]. However, compared to the traditional image pyramids, the pyramidal neural network is new in that nonlinear processing at pyramidal stages can be tuned, through learning, for specific recognition tasks. The pyramidal neural network also possesses several strengths of 2-D neural networks, including the integration of feature extraction and classification into

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 207–216, 2006.

a single structure, and the use of receptive fields to retain the 2-D spatial topology of image patterns. Furthermore, the PyraNet has a systematic connection scheme, which simplifies greatly the task of network design and enables generic training methods to be devised.

The paper is organized as follows. In the next section, we address the architectural aspects of the new PyraNet, and compare it to some related neural network models. In section 3, we derive a training method for the PyraNet that is based on the resilient back-propagation (RPROP) algorithm and the weight decay scheme. In section 4, we design a PyraNet to differentiate male and female facial patterns. Finally in section 5, we present some concluding remarks.

2 PyraNet Network Model

In this section, we first present the schematic structure and the mathematical model of the PyraNet. We then discuss the differences between the PyraNet and some related network models.

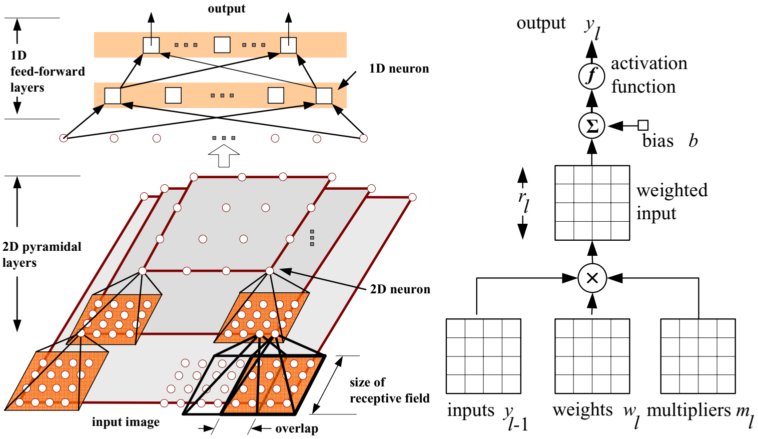
2.1 Network Structure

The PyraNet has a hierarchical multilayered structure as illustrated in Fig. 1a. A PyraNet contains two types of layers: 2-D pyramidal layers and 1-D feed-forward layers. The 2-D layers perform both 2-D feature extraction and data reduction, whereas the 1-D layers handle classification. The first pyramidal layer is connected to the input image, and followed by one or more pyramidal layers. The last pyramidal layer is connected to 1-D layers. With this cascading structure, the output of one layer becomes the input to the next layer.

A pyramidal layer is a grid of 2-D neurons, each of which is connected to a specific square region of the previous layer's output. A 2-D neuron, shown in Fig. 1b, first computes a weighted sum of the inputs in its receptive field; it then applies a nonlinear activation function to produce an output. The role of the 1-D feed-forward layers is to process the features produced by the 2-D pyramidal layers. Several 1-D layers may be needed in applications that involve complex decision boundaries. Nevertheless, it is expected that the use of pyramidal layers for 2-D feature extraction will simplify the task of feature classification by the 1-D layers. In theory, the 1-D layers can be constructed from any type of neurons, eg. radial basis function neurons or sigmoidal neurons. In this paper, we are interested in 1-D layers that consist of sigmoidal neurons (i.e. perceptrons). The outputs of the last 1-D layer are taken as the network outputs, which in visual recognition usually represent the categories of input patterns.

2.2 Mathematical Model

The notations for describing architectural aspects of the PyraNet are summarized in Table 1. The symbol l indicates the index of a network layer. For pyramidal layer $l = 1, 2, ..., L_p$, let r_l be the size of a receptive field, o_l be the horizontal or vertical overlap in pixels between two adjacent receptive fields. The difference g_l, i.e. $g_l = r_l - o_l$, is the gap between adjacent receptive fields. Let $f_l(.)$ be the activation function of layer l.



(a) network structure (b) 2-D neuron

Fig. 1. The pyramidal neural network architecture

Suppose we need to analyze an image pattern $\mathbf{X}$ of size $H_0 \times W_0$ pixels. The input image is partitioned into overlapping regions; each region consists of $r_1 \times r_1$ pixels and is considered as a receptive field to a neuron in layer 1. Each pixel in the input image is associated with an adjustable weight: let $w^1_{i,j}$ denote the weight for image pixel at position (i, j). Let $b^1_{i^*,j^*}$ be the bias of neuron (i^*, j^*) of layer 1. The output of the pyramidal neuron (i^*, j^*) in layer 1 is given by

$$y^1_{i^*,j^*} = f_1\Big(\sum_{i=i_{\text{low}}}^{i_{\text{high}}} \sum_{j=j_{\text{low}}}^{j_{\text{high}}} w^1_{i,j}\; x_{i,j}\; m^1_{i-i_{\text{low}}+1,j-j_{\text{low}}+1} + b^1_{i^*,j^*}\Big), \qquad (1)$$

where

- summation is defined over all positions (i, j) in the neuron's receptive field. That is, $i_{\text{low}} = (i^* - 1)g_1 + 1$, $i_{\text{high}} = (i^* - 1)g_1 + r_1$, $j_{\text{low}} = (j^* - 1)g_1 + 1$, and $j_{\text{high}} = (j^* - 1)g_1 + r_1$.
- $m^1_{\alpha,\beta}$, where $\alpha, \beta = 1, 2, ..., r_1$, denotes an entry in a fixed multiplier matrix of size $r_1 \times r_1$.

For other pyramidal layers, let $w^l_{i,j}$ be the synaptic weight associated with the input position (i, j) to layer l, and $b^l_{i^*,j^*}$ be the bias of neuron (i^*, j^*) in layer l. The output of the pyramidal neuron is computed in a similar way as (1):

$$y^l_{i^*,j^*} = f_l\Big(\sum_{i=i_{\text{low}}}^{i_{\text{high}}} \sum_{j=j_{\text{low}}}^{j_{\text{high}}} w^l_{i,j}\; y^{l-1}_{i,j}\; m^l_{i-i_{\text{low}}+1,j-j_{\text{low}}+1} + b^l_{i^*,j^*}\Big) \qquad (2)$$

where $i_{\text{low}} = (i^* - 1)g_l + 1$, $i_{\text{high}} = (i^* - 1)g_l + r_l$, $j_{\text{low}} = (j^* - 1)g_l + 1$, and $j_{\text{high}} = (j^* - 1)g_l + r_l$.

Table 1. Architectural notations for the PyraNet

Description	**Symbol**	**Note**
Input image size	N_0	$N_0 = H_0 \times W_0$ pixels
Numbers of layers	L_p, L_f, L	pyramidal, 1-D feed-forward, total
Layer index	l	$l = 1, ..., L_p, L_p + 1, ..., L_p + L_f$
Activation function of layer l	$f_l(.)$	$l = 1, 2, ..., L$
Size of a receptive field in pyramidal layer l	r_l	$l \leq L_p$
Receptive field overlap in layer l	o_l	$l \leq L_p$
Gap factor for pyramidal layer l	g_l	$g_l = r_l - o_l$
Multiplier matrix for pyramidal layer l	$\{m^l_{\alpha,\beta}\}$	$\alpha, \beta = 1, 2, ..., r_l$
Number of neurons in pyramidal layer l	N_l	$N_l = H_l \times W_l$, $H_l = \lfloor \frac{H_{l-1} - o_l}{g_l} \rfloor$, $W_l = \lfloor \frac{W_{l-1} - o_l}{g_l} \rfloor$
Weight associated with input position (i, j) to pyramidal layer l	$w^l_{i,j}$	$i = 1, ..., H_{l-1}$ $j = 1, ..., W_{l-1}$
Bias of neuron (i^*, j^*) in pyramidal layer l	$b^l_{i^*,j^*}$	$i^* = 1, ..., H_l,\ j^* = 1, ..., W_l$
Number of neurons in 1-D layer l	N_l	$l > L_p$
Weight from neuron q in 1-D layer $l-1$, to neuron r in layer l	$w^l_{q,r}$	$l > L_p, q = 1, ..., N_{l-1}$ and $r = 1, ..., N_l$
Bias of neuron r in 1-D layer l	b^l_r	$l > L_p$ and $r = 1, ..., N_l$

For layer l, the fixed multiplier $m^l_{\alpha,\beta}$, where $\alpha, \beta = 1, 2, ..., r_l$, is defined as:

$$m^l_{\alpha,\beta} = \frac{r_l^2}{(r_l + o_l)^2}[max(1, \alpha - o_l) - min(r_l - o_l, \alpha) + 1] \times [max(1, \beta - o_l) - min(r_l - o_l, \beta) + 1] \tag{3}$$

For example, the multiplier matrix for $r_l = 4$ and o_l = 2 is given as

$$m^l = \frac{4^2}{(4+2)^2}\begin{pmatrix} 1\,2\,2\,1 \\ 2\,4\,4\,2 \\ 2\,4\,4\,2 \\ 1\,2\,2\,1 \end{pmatrix} \tag{4}$$

The multiplier matrix resembles a lowpass filter. Combined with overlapping receptive fields, it improves the stability of 2-D neurons with respect to input shift. Note that the sizes of adjacent pyramidal layers are related as $H_l = \lfloor \frac{H_{l-1} - o_l}{g_l} \rfloor$ and $W_l = \lfloor \frac{W_{l-1} - o_l}{g_l} \rfloor$. For this reason, g_l is also called the pyramidal step of layer l.

The output $\{y^{L_p}_{i,j}\}$ of the last pyramidal layer is rearranged into a column vector, and used as input to the following 1-D feed-forward layer:

$$\{y^{L_p}_{i,j},\ \ i = 1, ..., H_{L_p}; j = 1, ..., W_{L_p}\} \rightarrow \{y^{L_p}_q,\ \ q = 1, ..., N_{L_p}\} \tag{5}$$

In this paper, the 2-D and 1-D formats for the last pyramidal layer are used interchangeably. For 1-D feed-forward layers, let $w^l_{q,r}$ be the synaptic weight from neuron q in layer $l-1$, to neuron r in layer l. Let b_r^l be the bias of neuron r in layer l; the output of the 1-D neuron is given by:

$$y_r^l = f_l(\sum_{q=1}^{N_{l-1}} w_{q,r}^l \, y_q^{l-1} + b_r^l) \tag{6}$$

The outputs of the neurons in the last layer, $\{y_r^L, r = 1, ..., N_L\}$, form the final network outputs.

2.3 Discussion of PyraNet Architecture

The PyraNet shares three properties with two-dimensional network models such as the convolutional neural networks [1]: (i) the network can process input image pixels directly; (ii) 2-D neurons are connected only to local regions; (iii) each pyramidal layer forms a compressed form of the outputs by the preceding layer. Note that 2-D layers in the PyraNet are not limited to dyadic image pyramids; depending on the application, each 2-D layer can have a different pyramidal step g_l.

The PyraNet differs from the convolutional neural networks in a number of aspects. Most importantly, the convolutional neural network are both based on weight-sharing, i.e. all neurons in a given convolution plane share the same set of weights or convolution mask. While weight sharing reduces the number of trainable network weights, it requires several planes or feature maps to be included in each convolution layer so that enough features can be extracted to support complex decision tasks. Furthermore, a feature map in convolutional network detects a feature at any input location. In contrast, each synaptic weight in the PyraNet is associated with a specific input position. Hence, a pyramidal neuron in the PyraNet reveals the presence of a feature (not limited to low-level features such as edges or lines) at a specific input location (i.e. the region the neuron is assigned to).

3 PyraNet Training

To complete the design of the proposed network, we present in this section a training algorithm for the PyraNet. Let $\{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^K\}$ be K training input samples, and $\{\mathbf{d}^1, \mathbf{d}^2, ..., \mathbf{d}^K\}$ be the desired output samples. The superscript k is used to indicate a sample in the training set. Our objective is to reduce iteratively the following mean-square-error between the actual and desired outputs:

$$E(\mathbf{w}) = \frac{1}{K \times N_L} \sum_{k=1}^{K} \sum_{r=1}^{N_L} |e_r^k|^2 \tag{7}$$

where e_r^k is the error in the qth output for the kth input sample, $e_r^k = y_r^{L,k} - d_r^k$, and $\mathbf{w}$ is a vector representing all weights and biases. Most algorithms for minimizing E require its gradient $\nabla\mathbf{E}$. Hence, our first step is to compute the error gradient for the PyraNet.

3.1 PyraNet Error Gradient Computation

For an input sample k, let $s^{l,k}_{i^*,j^*}$ be the weighted sum input to neuron (i^*, j^*) in pyramidal layer l. Let $s^{l,k}_r$ be the weighted sum input to neuron r in 1-D layer l. The error gradient is computed through error sensitivities, which are defined as the partial derivatives of the error E with respect to weighted sum input to individual neurons,

$$\text{for 2D neurons : } \quad \delta^{l,k}_{i^*,j^*} = \frac{\partial E}{\partial s^{l,k}_{i^*,j^*}}, \;\; l \leq L_p \tag{8}$$

$$\text{for 1D neurons : } \quad \delta^{l,k}_{r} = \frac{\partial E}{\partial s^{l,k}_{r}}, \;\; l > L_p \tag{9}$$

Using the chain rule of differentiation, we can express the error sensitivities as follows.

* **For the last 1-D layer**, where $r = 1, ..., N_L$:

$$\delta^{L,k}_r = \frac{2}{K \times N_L}\, e^k_r\, f'_L(s^{L,k}_r) \tag{10}$$

* **For other 1-D layers**, where $L_p < l < L$ and $r = 1, ..., N_l$:

$$\delta^{l,k}_r = f'_l(s^{l,k}_r) \sum_{q=1}^{N_{l+1}} \delta^{l+1,k}_q\, w^{l+1}_{r,q} \tag{11}$$

* **For the last pyramidal layer**, the error sensitivities $\{\delta^{L_p,k}_r\}$ can be calculated using (11) for $l = L_p$, but they must be rearranged into a 2-D grid:

$$\{\delta^{L_p,k}_r,\;\; r = 1, ..., N_{L_p}\} \rightarrow \{\delta^{L_p,k}_{i^*,j^*},\;\; i^* = 1, ..., H_{L_p}; j^* = 1, ..., W_{L_p}\} \tag{12}$$

* **For other pyramidal layers**, where $l < L_p$, $i^* = 1, ..., H_l$ and $j^* = 1, ..., W_l$:

$$\delta^{l,k}_{i^*,j^*} = f'_l(s^{l,k}_{i^*,j^*})\, w^{l+1}_{i^*,j^*} \times \sum_{i=i_{\text{low}}}^{i_{\text{high}}} \sum_{j=j_{\text{low}}}^{j_{\text{high}}} \delta^{l+1,k}_{i,j}\, m^{l+1}_{i^*-(i-1)g_{l+1},j^*-(j-1)g_{l+1}} \tag{13}$$

where $i_{\text{low}} = \lceil \frac{i^*-r_{l+1}}{g_{l+1}} \rceil + 1$, $i_{\text{high}} = \lfloor \frac{i^*-1}{g_{l+1}} \rfloor + 1$, $j_{\text{low}} = \lceil \frac{j^*-r_{l+1}}{g_{l+1}} \rceil + 1$, and $j_{\text{high}} = \lfloor \frac{j^*-1}{g_{l+1}} \rfloor + 1$.

Finally, we can calculate the error gradient.
* **For 1-D layers**, where $L_p < l \leq L$
◇ Weights $w^l_{q,r}$ where $q = 1, ..., N_{l-1}$ and $r = 1, ..., N_l$:

$$\frac{\partial E}{\partial w^l_{q,r}} = \sum_{k=1}^{K} \delta^{l,k}_r\, y^{l-1,k}_q \tag{14}$$

◇ Biases b^l_r where $r = 1, ..., N_l$:

$$\frac{\partial E}{\partial b^l_r} = \sum_{k=1}^{K} \delta^{l,k}_r \tag{15}$$

* **For pyramidal layers** where $l \leq L_p$
◇ Weights $w^l_{i,j}$ where $i = 1, ..., H_{l-1}$ and $j = 1, ..., W_{l-1}$:

$$\frac{\partial E}{\partial w^l_{i,j}} = \sum_{k=1}^{K} \{ y^{l-1,k}_{i,j} \times \sum_{i^*=i_{\text{low}}}^{i_{\text{high}}} \sum_{j^*=j_{\text{low}}}^{j_{\text{high}}} \delta^{l,k}_{i^*,j^*} \; m^l_{i-(i^*-1)g_l, j-(j^*-1)g_l} \} \tag{16}$$

In (16), $y^{0,k}_{i,j}$ refers to the input sample, and $i_{\text{low}} = \lceil \frac{i-r_l}{g_l} \rceil + 1$, $i_{\text{high}} = \lfloor \frac{i-1}{g_l} \rfloor + 1$, $j_{\text{low}} = \lceil \frac{j-r_l}{g_l} \rceil + 1$, and $j_{\text{high}} = \lfloor \frac{j-1}{g_l} \rfloor + 1$.

◇ Biases $b^l_{i^*,j^*}$ where $i^* = 1, ..., H_l$ and $j^* = 1, ..., W_l$:

$$\frac{\partial E}{\partial b^l_{i^*,j^*}} = \sum_{k=1}^{K} \delta^{l,k}_{i^*,j^*} \tag{17}$$

This completes the derivation of the error gradient for the PyraNet.

3.2 Resilient Back-Propagation and Weight Decay

We train the PyraNet using the resilient back-propagation (RPROP) algorithm, proposed by Riedmiller and Braun [5]. This algorithm, which is one of the fastest among first-order algorithms, discards information about the gradient magnitude and uses only the sign of the gradient. In the RPROP algorithm, an adaptive learning rate is assigned to each network weight. The learning rate is increased if the partial derivative of the error keeps the same sign, compared to the previous epoch; otherwise the learning rate is reduced. The weight update, $\Delta w_i(t) = w_i(t+1) - w_i(t)$, of the RPROP algorithm is given by

$$\Delta w_i(t) = -sign\{\frac{\partial E}{\partial w_i}(t)\} \Delta_i(t) \tag{18}$$

where $\Delta_i(t)$ is the adaptive learning rate at iteration t for network parameter w_i,

$$\Delta_i(t) = \begin{cases} \eta_{\text{inc}} \, \Delta_i(t-1), & \text{if } \frac{\partial E}{\partial w_i}(t) \, \frac{\partial E}{\partial w_i}(t-1) > 0 \\ \eta_{\text{dec}} \, \Delta_i(t-1), & \text{if } \frac{\partial E}{\partial w_i}(t) \, \frac{\partial E}{\partial w_i}(t-1) < 0 \\ \quad \Delta_i(t-1), & \text{otherwise} \end{cases} \tag{19}$$

and $\eta_{\text{inc}} > 1$ and $0 < \eta_{\text{dec}} < 1$ are two scalar parameters.

To improve generalization and noise tolerance of the trained network, we also adopt the weight decay approach. An extra term is added to the MSE to form the overall objective function:

$$E_{\text{o}} = E + \frac{\lambda}{P} \sum_{i=1}^{P} w_i^2 \tag{20}$$

where P is the total number of weights and biases, and λ is a small positive scalar.

4 Gender Classification Using PyraNet

Gender classification of facial images is an interesting problem in vision. It has several applications such as counting male and female customers entering a shop and presenting gender-relevant information to computer users. Humans use many visual heuristics to differentiate men and women's faces. For example, women usually have a lighter facial skin, a smaller and thinner nose, thinner and higher eyebrows, a plumper cheek, and a harder facial outline compared to men. In this paper, we are interested in machine learning approaches to gender classification. Examples of existing gender classification approaches include perceptron [6], support vector machines (SVM) [7], and linear discriminant analysis (LDA) [8]. In this section, we apply a PyraNet to classify a facial image into two classes: male or female.

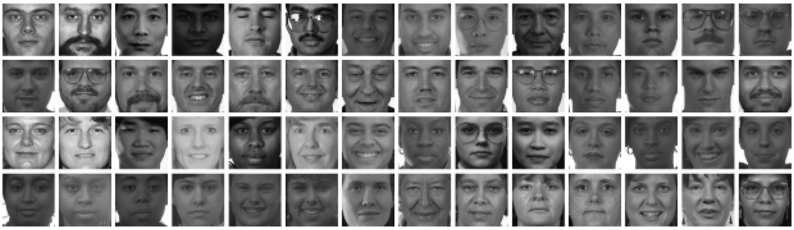


Fig. 2. Examples of male (rows 1-2) and female (rows 3-4) face patterns

For gender classification study, we use a standard and publicly available database - the FERET database [9]. This database consists of 14051 grayscale images of human faces, with views ranging from frontal to left and right profiles. Since gender classification mainly deals with frontal facial images, the entire FERET *frontal dataset*, also known as set fa, was used. This dataset has a total of 1762 images of 1010 subjects. The ground-truth (gender and face position) for about 90% of these images is provided as part of the 2003 Color FERET DVD[1]; the missing ground-truths were manually added by us. In this dataset, the face patterns include different ethnicities (Caucasian/ South Asian/East Asian/African), facial expressions (neutral/smiling), facial make-up (with/without glasses or beard), and lighting conditions (dark/normal). Examples of male and female face patterns are shown in Fig. 2. The ratio of male to female face patterns in this dataset is approximately 1.8 to 1. In our experiments, the extracted face patterns were histogram-equalized (similar lighting normalization was used in [7]), and then scaled to the range $[-1, 1]$.

We experimented with a number of networks and input image sizes. PyraNet1 has an input image size of 32×32 pixels and two 2-D layers with receptive field sizes $r_1 = 5$, $r_2 = 4$ and overlap sizes $o_1 = 2$ and $o_2 = 2$. PyraNet2 has an input image size of 30×30 pixels and two 2-D layers with $r_1 = 4$, $r_2 = 4$ and $o_1 = 2$ and $o_2 = 2$. Both PyraNet1 and PyraNet 2 have an output layer with one neuron and the hyperbolic tangent as the activation function: $f(\xi) = (e^{\xi} - e^{-\xi})/(e^{\xi} + e^{-\xi})$. PyraNet1

[1] Web site: http://www.itl.nist.gov/iad/humanid/colorferet/

Table 2. Performance comparison of gender classifiers

	PyraNet1	PyraNet2	CNN1	CNN2	k-NN1	k-NN2	SVM [7]
Maximum classification rate $\mathrm{CR}_{\max}(\%)$	96.3	96.2	89.8	89.3	92.5	87.1	96.6
95% confidence interval of $\mathrm{CR}_{\max}$	[95.4, 97.2]	[95.3, 97.1]	[88.4, 91.2]	[87.9, 90.7]	[91.3, 93.7]	[85.5, 88.7]	[95.8, 97.4]

has 1257 trainable parameters, whereas PyraNet2 has 1365 parameters. The PyraNets were trained to generate output of 1 and -1 for male and female patterns, respectively.

A five-fold cross validation was conducted: the dataset was divided into five subsets of equal sizes. For each fold, four subsets were used for designing the PyraNet and the remaining subset was used for testing. Of the data for designing the PyraNet, 9 tenths were used for training networks, and a tenth was used, as a verification set, for selecting the best network to be run on the test set. The classification rates on the test sets were averaged over the five folds.

For comparison purposes, we also implemented and tested two types of gender classifiers: the k-nearest neighbor (k-NN) classifier and the convolutional neural network (CNN) classifier, using the same dataset as for the PyraNet classifier. A k-NN classifier stores selected samples in its training set as prototypes. During testing, the class label of a new sample is determined (through majority-voting) based on the class labels of its k nearest prototypes. Since training the k-NN classifier is fast and requires little tuning from the designer, it is usually in pattern recognition as a comparison baseline. Two k-NN classifiers were tested: k-NN1 stores 100% of its training set and uses $k = 1$ nearest neighbor; k-NN2 stores 50% of its training set and uses $k = 5$ nearest neighbors.

We also compared with gender classifiers based on the convolutional neural network because this network architecture is closely related to the new PyraNet. Two convolutional network classifiers [2,1] were tested: CNN1 uses an input image size of 36×32 pixels and has 951 trainable parameters; CNN2 uses an input image size of 32×32 pixels and has 1853 trainable parameters. Both CNN1 and CNN2 have six layers: three convolutional layers, two sub-sampling layers, and one output layer. Note that a minor difference in the input image sizes of PyraNets and CNN1 is necessary to prevent clipping at border pixels.

The classification performance of different gender classifiers are shown in Table 2. The results were obtained on the receiver operating characteristics (ROC) curves of the classifiers where the classification rates are maximum. Evaluated on FERET dataset, PyraNet1 and PyraNet2 have CRs of 96.3% and 96.2% respectively whereas CNN1 and and CNN2 have CRs of 89.8% and 89.3% respectively. Both PyraNet1 and PyraNet2 have higher classification rates (CRs) compared with the k-NN and CNN classifiers. It is interesting to see that k-NN1 storing 100% of its training set has quite good performance (CR of 92.5%).

Using the standard FERET dataset, we can compare directly our technique and the SVM technique proposed by Moghaddam and Yang [7]. Moghaddam and Yang's SVM technique is one of the state-of-the-art techniques for gender classification, and it achieves a classification rate of 96.6% on the FERET frontal dataset. Our PyraNet gender classifiers have very similar classification rates as the SVM gender classifier. A key advantage of our approach is that the PyraNet gender classifiers have fewer than 1400 trainable parameters whereas the SVM gender classifier has over 75600 trainable parameters (about 20% of the training samples are used as support vectors).

In terms of computational complexity, to compute a pyramidal layer in the PyraNet when the input layer has a size of $h \times w$, the number of operations (additions, multiplications, and activation function evaluations) required is approximately $3(r/g)^2 \times h \times w$, where r is the receptive field's width, g is the gap factor, and $(r/g)^2$ is typically less than 10. In comparison, for the SVM with the RBF kernel, if there are k support vectors and each vector has a size of $h \times w$, the number of operations required is approximately $3k \times h \times w$; k is in the order of hundreds.

5 Conclusion

This paper presents a new pyramidal network with a hierarchical structure that can process image pixels directly. The new network is based on 2-D neurons that are connected to local regions of the image; these neurons are trained to extract 2-D features that have strong spatial dependency. We have derived a generic training algorithm for the PyraNet and applied the PyraNet in gender classification of facial images. Evaluated on the FERET dataset of 1762 images, the PyraNet gender classifier achieves a classification rate of 96.3%.

References

1. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324
2. Garcia, C., Delakis, M.: Convolutional face finder: A neural architecture for fast and robust face detection. IEEE Trans. on Pattern Analysis and Machine Intelligence **26**(11) (2004) 1408–1423
3. Fukushima, K.: Neocognitron: a hierarchical neural network capable of visual pattern recognition. Neural Networks **1**(2) (1988) 119–130
4. Gonzalez, R.C., Woods, R.E.: Digital image processing. Prentice Hall, New York (2002)
5. Riedmiller, M., Braun, H.: A direct adaptive method of faster backpropagation learning: The rprop algorithm. In: IEEE International Conference on Neural Networks, San Francisco (1993) 586–591
6. Gray, M., Lawrence, D.T., Golomb, B.A., Sejnowski, T.J.: A perceptron revealing the face of sex. Neural Computation **7**(6) (1995) 1160–1164
7. Moghaddam, B., Yang, M.H.: Learning gender with support faces. IEEE Trans. on Pattern Analysis and Machine Intelligence **24**(5) (2002) 707–711
8. Jain, A., Huang, J.: Integrating independent components and linear discriminant analysis for gender classification. In: IEEE International Conference on Automatic Face and Gesture Recognition. (2004) 159–163
9. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face-recognition algorithms. IEEE Trans. on Pattern Analysis and Machine Intelligence **22**(10) (2000) 1090–1104

A Novel Model for Gabor-Based Independent Radial Basis Function Neural Networks and Its Application to Face Recognition

GaoYun An and QiuQi Ruan

Institute of Information Science, Beijing Jiaotong University, Beijing, China, 100044
gaoyun_an@126.com, qqruan@center.njtu.edu.cn

Abstract. In this paper, a novel model for Gabor-based independent radial basis function (IRBF) neural network is proposed and applied to face recognition. In the new model, a bank of Gabor filters is first built to extract Gabor face representations characterized by selected frequency, locality and orientation to cope with various illuminations, facial expression and poses in face recognition. Then principal component analysis (PCA) is adopted to reduce the dimension of the extracted Gabor face representations for every face sample. At last, a new IRBF neural network is built to extract high-order statistical features of extracted Gabor face representations with lower dimension and to classify these extracted high-order statistical features. According to the experiments on the famous CAS-PEAL face database, our proposed approach could outperform ICA with architecture II (ICA2) and kernel PCA (KPCA) with standing testing sets proposed in the current release disk of the CAS-PEAL face database.

1 Introduction

Up to now, there have been many successful algorithms for face recognition. But there are still some outliers which will impact the performance of face recognition algorithms. These outliers are facial expression, illumination, pose, masking, occlusion etc. So how to make current algorithms robust to these outliers or how to develop some powerful classifiers is the main task for face recognition. Principal Component Analysis (PCA) [6], Fisher's Linear Discriminant (FLD) [7] and Independent Component Analysis (ICA) [4] are three basic algorithms for subspace analysis in face recognition. According to the three basic algorithms, some improved algorithms have also been proposed, like the work of Schölkopf etc. [3], Liu [1] and Yang etc. [5].

With the help of a possibly nonlinear map, Schölkopf etc. [3] extended PCA to a kernel PCA which could take advantage of arbitrary high-order statistical relationship among various input variables. Liu [1] combined the Gabor Wavelet and kernel PCA together to propose a new face recognition algorithm. Yang etc. [5] combined kernel PCA and ICA to propose an alternative KICA algorithm for face recognition.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 217 – 226, 2006.

From another point of view, algorithms proposed in [1] and [3]-[7] are just for the feature extraction stage in an identification system. Some powerful classifiers are expected to classify these extracted features. Radial basis function (RBF) neural network is an ideal choice due to its nonlinear classifying property. Meng etc. [2] has successfully tried to use RBF neural network to classify features extracted by FLD.

According to the above analysis, a novel model for Gabor-based independent radial basis function (IRBF) neural network is proposed and applied to face recognition. In the new model, a bank of Gabor filters is first built to extract Gabor face representations characterized by selected frequency, locality and orientation to cope with various illuminations, facial expression and poses in face recognition. Then PCA is adopted to reduce the dimension of the extracted Gabor face representations for every face sample. At last, a new IRBF neural network is built to extract high-order statistical features of extracted Gabor face representations with lower dimension and to classify these extracted high-order statistical features. The detail about the new algorithm will be discussed in the next section.

In the technical report (2004) [10], Delac et al. have confirmed that ICA with architecture II (ICA2) proposed by Bartlett et al.[4] could outperform PCA and FLD on a large scale face database. So in our experiments, we just compare our proposed approach with kernel PCA (KPCA) employing polynomial kernels and ICA2. A newly built and famous CAS-PEAL [9] face database is chosen to confirm the validity of the new algorithm. The current release of CAS-PEAL face database contains 30864 face samples of 1040 subjects. According to the experiments on the CAS-PEAL face database, our proposed approach could outperform ICA2 and KPCA with standing testing sets proposed in [9].

2 Gabor-Based IRBF Neural Networks

Given the training set $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, where N is the number of samples and $\mathbf{x}_i \in \mathbb{R}^{n\times n}$ is the gray level distribution of the ith face image, where $i = 1, \cdots, N$. The whole algorithm of face recognition by Gabor-based IRBF neural network could be discussed as follows.

2.1 Multiresolution Gabor Face Representations

In order to extract multiresolution Gabor face representations, a bank of Gabor filters is defined as:

$$G_{\mu,\upsilon}(\alpha) = \frac{\left\|k_{\mu,\upsilon}\right\|^2}{\sigma^2} e^{\frac{-\left\|k_{\mu,\upsilon}\right\|^2 \left\|\alpha\right\|^2}{2\sigma^2}} [e^{i\vec{k}_{\mu,\upsilon}\alpha} - e^{-\sigma^2/2}] \tag{1}$$

where μ defines the orientation of Gabor filters, υ defines the scale of Gabor filters to determine the center in the frequency domain, $\alpha = (x, y)$, $\left\|\bullet\right\|$ is the norm operator, and $k_{\mu,\upsilon}$ is defined as:

$$k_{\mu,\upsilon} = k_{\upsilon} e^{i\phi_\mu} \tag{2}$$

where $k_\upsilon = k_{\max} / f^\upsilon$ and $\phi_\mu = \pi\mu/8$. $k_{\max}$ is the maximum frequency, and f is the spacing factor between filters in the frequency domain.

Here the scale factor υ is chosen as $\upsilon \in \{0,\cdots,4\}$, the orientation factor μ is chosen as $\mu \in \{0,\cdots,7\}$, $\sigma = 2\pi$, $k_{\max} = \pi/2$ and $f = \sqrt{2}$. So a bank containing 40 Gabor filters is built as $\{G_{\mu,\upsilon}(\alpha) : \mu \in \{0,\cdots,7\}, \upsilon \in \{0,\cdots,4\}\}$.

Now the bank of Gabor filters is adopted to extract multiresolution Gabor face representations for every sample image in training set X as:

$$\mathbf{y}_{\mu,\upsilon}^{(i)} = \mathbf{x}_i * G_{\mu,\upsilon}(\alpha) \,,\; i = 1,\cdots,N \tag{3}$$

where $*$ is the convolution operator and $\mathbf{y}_{\mu,\upsilon}^{(i)}$ is the Gabor face representation of the ith sample images corresponding to the Gabor filter with orientation μ and scale υ .

In order to facilitate the computation of the following PCA algorithm, a Gabor feature vector corresponding to the ith sample image is defined as:

$$\mathbf{u}^{(i)} = [V(D_\rho(\mathbf{y}_{0,0}^{(i)})) \quad V(D_\rho(\mathbf{y}_{0,1}^{(i)})) \quad \cdots \quad V(D_\rho(\mathbf{y}_{4,7}^{(i)}))]^T \tag{4}$$

where $D_\rho(\mathbf{y})$ stands for downsampling the Gabor face representation $\mathbf{y}$ with a factor ρ on the two directions of the image respectively and $V(\bullet)$ stands for formatting a matrix to a row vector by concatenating its rows or columns.

Then a training matrix $\mathbf{U}$ is defined as $\mathbf{U} = [\mathbf{u}^{(1)} \quad \mathbf{u}^{(2)} \quad \cdots \quad \mathbf{u}^{(N)}]$ to train the following PCA algorithm for dimension reduction.

2.2 Principal Component Analysis

Before transferring every sample in training matrix $\mathbf{U}$ into the new IRBF neural networks for classification, we should reduce the dimension of every sample with PCA. In this step, the projection matrix $\mathbf{W}_{pca}$ is calculated as:

$$\begin{aligned} \mathbf{W}_{pca} &= \arg\max_{\mathbf{W}} \left| \mathbf{W}^T \aleph \mathbf{W} \right| \\ &= [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \cdots \quad \mathbf{w}_m] \end{aligned} \tag{5}$$

where m is the dimension of the PCA feature space. Matrix $\aleph$ is the total scatter matrix, and is calculated as:

$$\aleph = \sum_{i=1}^{N} (\mathbf{u}_i - \boldsymbol{\mu})(\mathbf{u}_i - \boldsymbol{\mu})^T \tag{6}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{40n^2/\rho^2}$ is the mean vector of all samples in training matrix $\mathbf{U}$.

All the samples in training matrix **U** are projected into the PCA feature space as:

$$\mathbf{z}_i = \mathbf{W}_{pca}^T(\mathbf{u}_i - \boldsymbol{\mu}) \tag{7}$$

where $\mathbf{z}_i$ is the representation of sample $\mathbf{u}_i$ in the PCA feature space. And all the $\mathbf{z}_i$, $i = 1, \cdots, N$, will been transferred into the IRBF neural network as input samples.

2.3 Independent Radial Basis Function Neural Networks

A classical RBF neural network is formed by three layers: input layer, hidden layer and output layer. It directly projects the input samples into a high dimension feature space through some radial basis functions $\varphi_i(\bullet)$, and does not take account of the high-order statistical relationship among variables of input samples. As known, the high-order statistical relationship does play an important part in pattern recognition (classification) area. So in order to take advantage of the high-order statistical relationship among variables, we proposed the independent radial basis function (IRBF) neural network.

As shown in Fig.1, the IRBF neural network contains four layers: input layer, unmixing layer, hidden layer and output layer. The input layer just transfers the input samples $\mathbf{z} = [z_1, z_2, \cdots, z_m]^T$ to the unmixing layer.

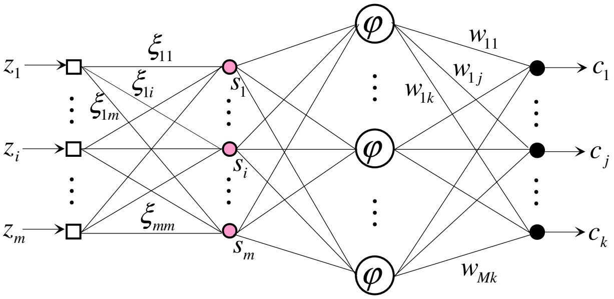


Fig. 1. The main structure of IRBF neural networks

The unmixing layer extracts the high-order statistical relationship among variables of the input samples transferred from input layer as follows:

$$s_i = f_i\left(\sum\nolimits_{p=1}^{m} \xi_{pi} z_p\right),\ i = 1, \cdots, m \tag{8}$$

where s_i, $i = 1, \cdots, m$ is statistical independent, and ξ_{pi} is one component of unmixing matrix $\Xi_{m\times m}$. Function $f_i(\vartheta)$ is an invertible squashing function, mapping real numbers into the [0, 1] interval. Here, we chose $f_i(\vartheta) = 1/(1+e^{-\vartheta})$.

In order to achieve the independence among s_i, $i = 1,\cdots,m$, an information maximization approach [11] to blind separation and blind deconvolution is used as follows. The relationship between joint entropy $H(\mathbf{s})$ and mutual information $I(\mathbf{s})$ is defined as:

$$H(s_1,\cdots,s_m) = H(s_1)+,\cdots,+H(s_m) - I(s_1,\cdots,s_m) \tag{9}$$

where $\mathbf{s} = [s_1,\cdots,s_m]^T$.

Since independent components have zero mutual information, as proposed by [11] the objective of independence among s_i, $i = 1,\cdots,m$ could be achieved by maximizing the joint entropy $H(\mathbf{s})$:

$$\begin{aligned}\Xi_{opt} = \arg\max_{\Xi} H(f_1(\sum_{p=1}^{m}\xi_{p1}z_p))+,\cdots,+H(f_m(\sum_{p=1}^{m}\xi_{pm}z_p)) \\ - I(f_1(\sum_{p=1}^{m}\xi_{p1}z_p),\cdots,f_m(\sum_{p=1}^{m}\xi_{pm}z_p))\end{aligned} \tag{10}$$

The optimization of unmixing matrix Ξ_{opt} could be calculated through the following gradient update rule [4]:

$$\Delta\Xi \propto \nabla_{\Xi} H(\mathbf{s}) = (\Xi^T)^{-1} + E(\mathbf{s}'\mathbf{z}^T) \tag{11}$$

where $\mathbf{s}' = [s_1',\cdots,s_m']^T$ and $s_i' = f_1''(\sum_{p=1}^{m}\xi_{pi}z_p)/f_i'(\sum_{p=1}^{m}\xi_{pi}z_p)$. $E(\bullet)$ stands for calculating the expected value.

After getting the optimal unmixing matrix Ξ_{opt}, for all the input samples $\mathbf{z}^{(i)}$ the new feature vector $\mathbf{s}^{(i)} = [s_1,\cdots,s_m]^T$ which reflects the high-order statistical relationship could be calculated by Eq. (8).

Then these new feature vectors $\mathbf{s}^{(i)}$ are mapped into a high dimension feature space through the radial basis function $\varphi(\bullet)$ of the hidden layer. In our proposed IRBF neural network, the number of nodes of hidden layer is equal to the number of input training samples, and the radial basis function is defined as:

$$\varphi_i(\mathbf{s}) = \psi(\|\mathbf{s} - \mathbf{t}_i\|) , \ i = 1,2,\cdots,N \tag{12}$$

where $\mathbf{t}_i$ is the center and is chosen as $\mathbf{t}_i = \mathbf{s}_i$ in this paper. Function $\psi(\bullet)$ chooses multiquadrics function.

Now the output of the *j*th output node of IRBF neural network is defined as:

$$\Gamma_j(\mathbf{s}) = \sum_{i=1}^{N} w_{ij}\psi(\mathbf{s},\mathbf{s}_i) = \sum_{i=1}^{N} w_{ij}\psi(\|\mathbf{s} - \mathbf{s}_i\|) \tag{13}$$

At last, the weighted matrix **W** between hidden layer and output layer is calculated through the following optimization problem:

$$\begin{aligned}\mathbf{W}_{opt} &= \arg\max_{\mathbf{W}} E(\Gamma) \\ &= \arg\max_{\mathbf{W}} \sum_{i=1}^{N} \sum_{j=1}^{k} (c_{ij} - \sum_{p=1}^{N} w_{pj} \psi(\|\mathbf{s}_i - \mathbf{t}_p\|))^2\end{aligned} \tag{14}$$

The solution to Eq. (14) could be calculated by Eq. (15), and the detail about the calculating procedure could be referred to [8].

$$\mathbf{W}_{opt} = (\Psi^T \Psi)^{-1} \Psi^T \mathbf{C} \tag{15}$$

where **C** is a $N \times k$ matrix of target output and k is the number of classes. Matrix Ψ is :

$$\Psi = \begin{bmatrix} \psi(\mathbf{s}_1,\mathbf{s}_1) & \psi(\mathbf{s}_1,\mathbf{s}_2) & \cdots & \psi(\mathbf{s}_1,\mathbf{s}_N) \\ \psi(\mathbf{s}_2,\mathbf{s}_1) & \psi(\mathbf{s}_2,\mathbf{s}_2) & \cdots & \psi(\mathbf{s}_2,\mathbf{s}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \psi(\mathbf{s}_N,\mathbf{s}_1) & \psi(\mathbf{s}_N,\mathbf{s}_2) & \cdots & \psi(\mathbf{s}_N,\mathbf{s}_N) \end{bmatrix} \tag{16}$$

2.4 Summary

The training procedure of the new IRBF neural network contains two steps: first, the unmixing matrix Ξ_{opt} should be adjusted by an information maximization approach; second, the weighted matrix $\mathbf{W}_{opt}$ between hidden layer and output layer should be tuned with Eq. (15).

At last, the whole algorithm of face recognition by Gabor-based IRBF neural network could be summarized as:

- First, a bank of Gabor filters should be built for multiresolution analysis. The Gabor face representations are then extracted with the bank of Gabor filters.
- Second, PCA is adopted to reduce the dimension of every Gabor face representation of every sample.
- Third, IRBF neural network is built to extract high-order statistical features of extracted Gabor face representations with lower dimension and to classify these extracted high-order statistical features.

The new algorithm of face recognition by Gabor-based IRBF neural network has the following advantages:

- Due to a bank of Gabor filters are used, the new algorithm is robust to illumination, facial expression and pose in face recognition.
- Due to a non-linear IRBF neural network is used, the new algorithm could take advantage of the high-order statistical features of every sample and classify various faces more efficiently.

3 Experimental Results

In our experiments, the famous CAS-PEAL face database built in 2004 is chosen to confirm the validity of various algorithms. The current release of CAS-PEAL face database contains 30864 images of 1040 subjects. They are with varying pose, expression, accessory and lighting. For each subject, 9 cameras spaced equally in a horizontal semicircular shelf are used to simultaneously capture images across different poses in one shot. Each subject is also asked to look up and down to capture 18 images in another two shots. 5 kinds of expressions, 6 kinds of accessories (3 pairs of glasses, and 3 caps), and 15 lighting directions are also considered. Detail about the CAS-PEAL face database may refer to [9] and the contents of current release of this face database are shown in Table 1(copied from [9]).

In the following experiments, all the face images are rotated, resized and cropped to 64x64 with 256 gray levels according to the coordinates of two eyes given in the current release of CAS-PEAL face database. The training set contains all the samples in the gallery set proposed in [9] and other 991 samples randomly selected from the face database, so total 2031samples are contained in the training set. Six probe sets corresponding to the six subsets in the frontal subsets: expression, lighting, accessory, background, distance and aging as described in Table 1 are chosen to test different algorithms. All the six probe sets will be referred as *Accessory, Aging, Background, Distance, Expression,* and *Lighting* in the following discussions, table and figures.

Table 1. The contents of current release of CAS-PEAL face database[9]

		Variations	Subjects	Images
Frontal	Normal	1	1040	1040
	Expression	5	377	1884
	Lighting	>=9	233	2450
	Accessory	6	438	2616
	Background	2-4	297	651
	Distance	1-2	296	324
	Aging	1	66	66
Pose		21(3*7)	1040	21832

In the technical report (2004) [10], Delac etc. have confirmed that ICA2 could outperform PCA and FLD on a large scale face database. So in our experiment, we just compare our proposed approach with KPCA employing polynomial kernels and ICA2 with four famous distances (*L1, L2, Cos* and *Md*) as similarity measurement for nearest neighbor classifier in face recognition. The dimension of the reduced feature space for our approach, KPCA and ICA2 is 300. And the four distances are:

$$L1: D_{L_1}(\mathbf{x},\mathbf{y}) = \sum_i |x_i - y_i| \tag{17}$$

$$L2: D_{L_2}(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y}) \tag{18}$$

$$Cos : D_{\cos}(\mathbf{x},\mathbf{y}) = -\mathbf{x}^T\mathbf{y} / \|\mathbf{x}\|\|\mathbf{y}\| \tag{19}$$

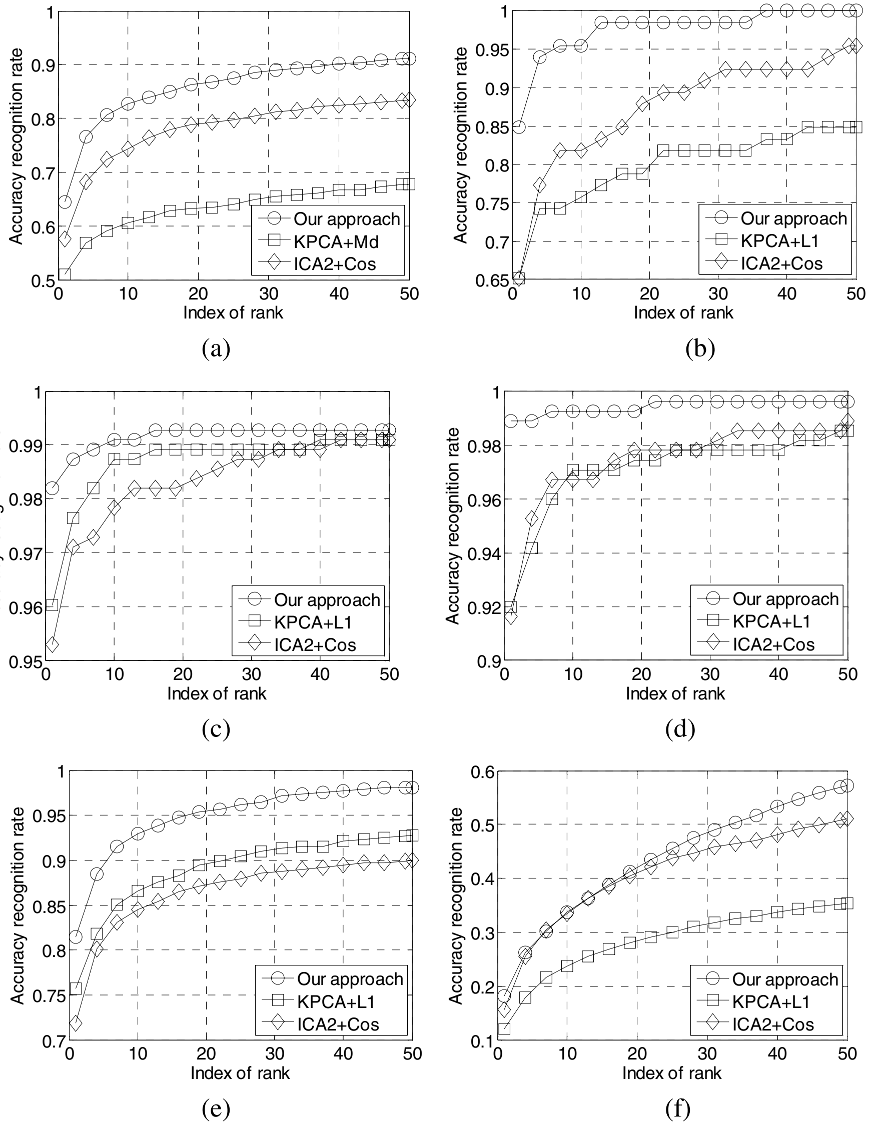
$$Md : D_{Md}(\mathbf{x},\mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \sum{}^{-1}(\mathbf{x} - \mathbf{y}) \tag{20}$$

Table 2 illustrates the accuracy recognition rate at rank 1 of KPCA, ICA2 and our proposed approach respectively. If the max accuracy recognition rates of various algorithms are considered, the accuracy recognition rate of our proposed approach is 13.5% (353/2616) and 6.8% (178/2616) higher than that of KPCA and ICA2 for the *Accessory* testing set; the accuracy recognition rate of our proposed approach is 19.7% (13/66) and 19.7% (13/66) higher than that of KPCA and ICA2 for the *Aging* testing set; the accuracy recognition rate of our proposed approach is 2.1% (14/651) and 2.9% (19/651) higher than that of KPCA and ICA2 for the *Background* testing set; the accuracy recognition rate of our proposed approach is 6.9% (22/324) and 7.3% (24/324) higher than that of KPCA and ICA2 for the *Distance* testing set; the accuracy recognition rate of our proposed approach is 5.8% (109/1884) and 9.6% (181/1884) higher than that of KPCA and ICA2 for the *Expression* testing set; and the accuracy recognition rate of our proposed approach is 6.2% (152/2450) and 2.5% (61/2450) higher than that of KPCA and ICA2 for the *Lighting* testing set. So it is clear that our proposed approach outperforms KPCA and ICA2 for all the six standing testing conditions. That is also confirmed that our proposed approach is more robust to accessory, aging, background, distance, facial expression and illumination than KPCA and ICA2. Fig. 2 also illustrates the accuracy recognition rate at rank 1 – 50 for the six testing conditions with the similarity measurement which has achieved the max accuracy recognition rate in Table 2. If the accuracy recognition rate at rank 10 is adopted, that of our proposed

Table 2. The accuracy recognition rate (%) at rank 1 of KPCA, ICA2 and our proposed approach

	L1	*L2*	*Cos*	*Md*	*RBF*	*L1*	*L2*	*Cos*	*Md*	*RBF*
			Accessory					*Aging*		
ICA2	44.8	49.1	**57.6**	50	—	30.3	34.9	**65.2**	36.4	—
KPCA	48.8	37.7	35.5	**50.9**	—	**65.2**	50	22.7	31.8	—
Our Approach	—	—	—	—	**64.4**	—	—	—	—	**84.9**
			Background					*Distance*		
ICA2	86.1	88.1	**95.3**	88.6	—	76.4	78.9	**91.6**	80.7	—
KPCA	**96.1**	92.6	84.6	87.7	—	**92**	90.6	72	78.2	—
Our Approach	—	—	—	—	**98.2**	—	—	—	—	**98.9**
			Expression					*Lighting*		
ICA2	56.6	63.3	**71.9**	63.6	—	6.2	8.3	**15.7**	8.8	—
KPCA	**75.7**	70.8	63.3	62	—	**12**	6.4	5.9	8.8	—
Our Approach	—	—	—	—	**81.5**	—	—	—	—	**18.2**

approach may reach 82.7% (2163/2616) for the *Accessory* testing set, 95.5% (63/66) for the *Aging* testing set, 99.1% (645/651) for the *Background* testing set, 99.3% (322/324) for the *Distance* testing set, 92.9% (1750/1884) for the *Expression* testing set and 33.7% (826/2450) for the *Lighting* testing set.



(a) (b)

(c) (d)

(e) (f)

Fig. 2. The accuracy recognition rates at rank 1 - 50 of KPCA, ICA2 and our proposed approach with the similarity measurement which has achieved the max accuracy recognition rate in Table 2. In the figure, (a) is for *Accessory* set, (b) is for *Aging* set, (c) is for *Background* set, (d) is for *Distance* set, (e) is for *Expression* set and (f) is for *Lighting* set.

4 Conclusions

In this paper, a novel model for Gabor-based independent radial basis function (IRBF) neural network is proposed and applied to face recognition. According to the experiments on the famous CAS-PEAL face database, our proposed approach could outperform ICA2 and KPCA with standing testing sets proposed in [9]. That is also confirmed that our proposed approach is more robust to accessory, aging, background, distance, facial expression and illumination than KPCA and ICA2 in face recognition.

Acknowledgement

This work was supported by the National Nature Science Foundation of China (60472033), by the Doctoral Program Foundation of the Ministry of Education of China (20030004023), and by the 973 Program of China (2004CB318005).

Portions of the research in this paper use the CAS-PEAL face database collected under the sponsor of the Chinese National Hi-Tech Program and ISVISION Tech. Co. Ltd., and the authors will also thank the anonymous reviewers.

References

1. Liu, C.J.: Gabor-based kernel PCA with fractional power polynomial models for face recognition. IEEE Trans. PAMI. vol. 26 (2004) 572-581
2. Meng, J.E., Wu, S., Lu, J., Hock, L.T.: Face recognition with radial basis function (RBF) neural networks. IEEE Trans. Neural Networks. vol. 13 (2002) 697-710
3. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation. vol. 10 (1998) 1299-1319
4. Bartlett, M.S., Movellan, J.R., Sejnowski, T.J.: Face recognition by independent component analysis. IEEE Trans. Neural Networks. vol. 13 (2002) 1450–1464
5. Yang, J., Gao, X., Zhang, D., Yang, J.: Kernel ICA: An alternative formulation and its application to face recognition. Pattern Recognition. vol. 38 (2005) 1784-1787
6. Turk, M., Pentland, A.: Eigenfaces for Recognition. Cognitive Neuroscience, vol. 3 (1991) 71-86
7. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Trans. PAMI. vol. 19 (1997) 711–720
8. Simon Haykin: Neural Networks: A Comprehensive Foundation, 2nd Edition. Pearson Education USA (1999)
9. Gao, W., Cao, B., Shan, S., Zhou, D., Zhang, X., Zhao, D.: The CAS-PEAL large-scale Chinese face database and evaluation protocols. Technical Report No. JDL_TR_04_FR_001, Joint Research & Development Laboratory, CAS, 2004
10. Delac, K., Grgic, M., Grgic, S.: Independent Comparative Study of PCA, ICA, and LDA on the FERET Data Set. Technical Report, University of Zagreb, FER (2004) www.face-rec.org/algorithms/ Comparisons/FER-VCL-TR-2004-03.pdf
11. Bell, A.J., Sejnowski, T.J.: An information-maximization approach to blind separation and blind deconvolution. Neural Computing. vol. 7 (1995) 1129–1159

Generalized PCA Face Recognition by Image Correction and Bit Feature Fusion

Huiyuan Wang, Yan Leng, Zengfeng Wang, and Xiaojuan Wu

School of Information Science and Engineering, Shandong University,
27 Shanda Nanlu, Jinan, Shandong 250100, China
{hywang, lyansdu, zengfengwang, xiaojwu}@sdu.edu.cn

Abstract. In this paper, two approaches to improve the illumination robustness of the face recognition algorithms are presented, that is, Symmetrical Image Correction (SIC) and Bit-Plan Feature Fusion (BPFF). SIC can reduce bright speckles and shadows caused by over lighting. BPFF constructs a new virtual face with Bit-Plan information of face images. Generalized PCA is then applied to the virtual faces to achieve face recognition. Experiments show that, the proposed combined method can reduce the sensitivity of face recognition to illuminations using fewer projection vectors than the compared approaches.

Keywords: Face recognition; Image correction; Bit-plane; Feature fusion; Generalized PCA.

1 Introduction

In face recognition, lighting can pollute a face image with big areas of bright speckles and shadows. It can cause great differences between face image matrices of the same person. These differences are inner differences, but usually, they can exceed faces' between-class differences [1]. So the recognition rate of many recognition algorithms decreases significantly with the introduction of lighting.

The preprocessing of face images is a necessary initial procedure of face recognition. The effect of preprocessing can greatly influence the succeeding recognition stages. Currently, face normalization is frequently used in face recognition [2]. But only normalization is far from enough because it doesn't do any essential correction to bright speckles and shadows. Takeshi Shakunaga presented a natural image correction algorithm that can weaken shadows caused by lighting [3]. But that algorithm is relatively complicated and has strong dependence on the number of training samples.

In this paper, we propose a novel image preprocessing algorithm—Symmetrical Image Correction (SIC) that is easy to be implemented. With SIC, we can reduce bright speckles and shadows in face images. Eigenface methods based on PCA have widely attracted researchers' attention due to its easy computation and realization. Generalized PCA based on bit-plan feature fusion proposed in this paper is an extended PCA algorithm. Hereinafter we call it BGPCA (Bit Generalized PCA) for short.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 227 – 235, 2006.

Classical face recognition algorithms such as eigenfaces and fisherface, extract features by projecting face images onto a given feature space. These algorithms do not usually take into account enhancing the discriminability of the original face images. While the Bit-Plan Feature Fusion algorithm proposed in this paper is to extract and fuse the bit-planes of the original face images, it keeps discriminant information to the greatest extent, and at the same time, reduce the influence of illumination and expression.

Specifically, the algorithm sets different weights to different bit-planes according to their contributions to recognition to increase the discriminability of the face samples.

In the experiments, SIC and BGPCA are combined to perform face recognition tasks. Experimental results show that, the integrated method can decrease the sensitivity of face recognition to illuminations. Furthermore, when the number of projection vectors is comparatively small, the proposed method can still achieve a good recognition rate.

2 Symmetrical Image Correction (SIC)

With one side light on, half of the face would be over-lighted and the other half of it would be over-darked. SIC tends to balance the difference. Given N gray images of a person with a gray value range of [0,255], a size of T×T, we perform histogram equalization on the images and denote the output by $\left\{x_j \mid j = 1,2,\cdots,N\right\}$. The correction result by SIC is:

$$x_j^*(m,n) = \left(1 - \left(x_j(m,n) - \bar{x}\right)/255\right) * x_j(m,n) \tag{1}$$

where $x_j(m,n)$ denotes the pixel of image x_j in row m and column n. And

$$\bar{x} = \frac{1}{2}\left(x_j(m,n) + x_j(m,T+1-n)\right) \tag{2}$$

In order to unify the gray value range, let us normalize the image as

$$X_j^*(m,n) = \frac{x_{ji}^*(m,n) - \min\left(x_j^*\right)}{\max\left(x_j^*\right) - \min\left(x_j^*\right)} \times 255 \tag{3}$$

For areas in a face image that are too bright, their values of $\left(x_j(m,n) - \bar{x}\right)/255$ are relatively large and positive; while for areas that are too dark, the absolute values of their $\left(x_j(m,n) - \bar{x}\right)/255$ are relatively large but negative. Thus, for areas that are too bright, their values of $1 - \left(x_j(m,n) - \bar{x}\right)/255$ are smaller than 1, while for areas that are too dark, $1 - \left(x_j(m,n) - \bar{x}\right)/255$ are bigger than 1. After correction, too bright areas would become relatively darker, and too dark areas would become relatively brighter. As a whole, the gray value distribution of the whole image tends to be even.

Above theoretical analysis indicates that SIC can reduce bright speckles and shadows introduced in an image by lighting. Fig.1 gives two triplets of image-correction examples. In each triplet, the left one represents an original face image, the center one is the histogram-equalization result of the left one, and the right one is the correction result by SIC on the center one. It is clear that, after SIC, the gray value distribution of the whole image tends to be even. Thereby, SIC decreases faces' within-class difference and makes samples of the same class close to the class center, thus makes samples of different classes to be more discriminant.

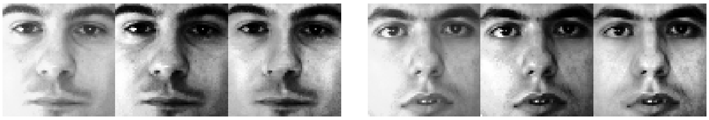


Fig. 1. Two triplets of image-correction example

3 Bit Generalized PCA (BGPCA)

3.1 Bit-Plane Information and Feature Fusion

Suppose that each pixel in a face image is represented by 8 bits, then a face image can be decomposed into 8 bit-planes.

Fig.2 shows a face image and its 8 bit-planes. In Fig.2, the image on the top is an original face image, the 8 images on the second row, from left to right, correspond to the bit planes 0 to 7 of the original image. Fig.2 shows the conventional bit-plane structure, i.e., the higher-order bits (especially the top four) contain the majority of the visually significant data, namely outline features, while the other bit planes contribute to more subtle details in the image.

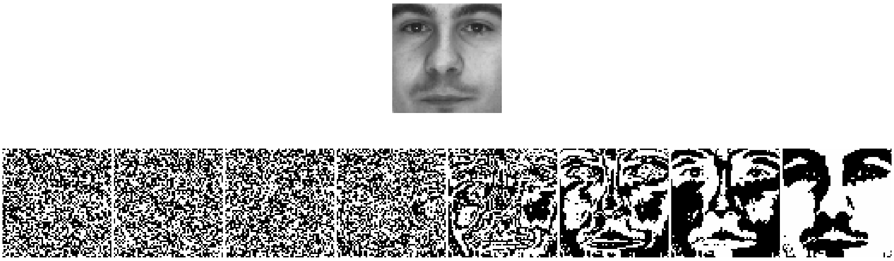


Fig. 2. An original face image and its 8 bit-planes

In our experiments, the images are preprocessed by histogram equalization. We found that after histogram equalization, the bit-plane structure is changed, as is shown in Fig.3. In order to confirm the universal applicability of this change, we have conducted observations on a wide variety of face images and non-face images. We extracted bit-planes from images before histogram equalization and after histogram equalization respectively, and then observed their bit-plane structures. Observation results indicate that before histogram equalization, the bit-plane structure obeys the same rule as in Fig.2, while for images after histogram equalization, their bit-plane structure follows a rule as is shown in Fig.3: bit–planes 0,1,5,6,7 include most of the outline features, and bit-planes 2,3,4 offer more subtle texture features in the image. The theoretical analysis to above observation is out of the range of this paper's discussion. It is to be discussed in a later paper.

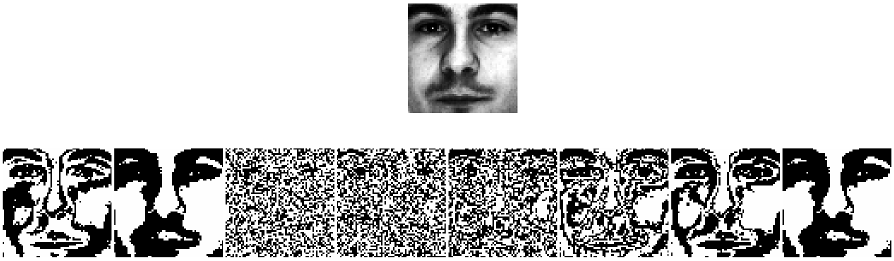


Fig. 3. A face image after histogram equalization and its 8 bit-planes

We know that although PCA takes into account all differences between images, it doesn't care whether the differences are caused by lighting, background, or the inner differences of faces. For several images of the same person, due to lighting, expression etc., the within-class differences can considerably exceed the between-class differences [1]. Researches indicate that the recognition rate of PCA decreases quickly with the introduction of variant illuminations and expressions. So it is important to reduce the within-class differences. To do so, let us set a common class-mark for each face image that belongs to the same class, which acts as an outline feature, and then this feature is fused with the weighted texture-feature of each face itself. Thus, since face images of the same class have the common class-mark, each sample of the class would cluster around the class-center, and their differences rest with the texture details. If the texture features are weighted according to their contributions to recognition, then, generally, for samples constructed by feature fusion, their within-class differences will decrease, and their between-class differences will increase.

We think that, the gray value of original face images reflects within-class differences overmuch due to the influence of illumination and express, thus it goes against classification. When we extract the bit-planes of face images, it is found that each bit-plane disperses the influence of illumination and expression. Furthermore, since each bit-plane is weighted differently, it can decrease the influence of illumination and expression to a certain extent. After feature fusion, the outline feature and the texture feature would be complementary, thus guarantee the reservation of discriminant information. In this way, the fusion algorithm can hopefully improve the precision of recognition.

The feature fusion procedure is stated as follows:

For training stage, let x_n^l denote the nth image (after histogram equalization) of the lth person($n=1,\cdots,N$; $l=1,\cdots,L$), perform bit-plane extraction on x_n^l to get its 8 bit-planes, and mark them as B_{nm}^l $(m=0,1,\cdots,7)$

Let

$$S_n^l = \sum_{m=0,1,5,6,7} \alpha_m B_{nm}^l \text{ and } T_n^l = \sum_{m=2,3,4} \beta_m B_{nm}^l \tag{4}$$

then

$$S^l = \frac{1}{N}\sum_{n=1}^{N} S_n^l \tag{5}$$

And

$$F_n^l = S^l + j\,T_n^l \tag{6}$$

Let $\bar{x}$ denotes the mean image of lth person, get the 8 bit-planes of $\bar{x}$, and mark them as B_m $(m=0,1,\cdots,7)$.

Let

$$S = \sum_{m=0,1,5,6,7} \alpha_m B_m \text{ and } T = \sum_{m=2,3,4} \beta_m B_m \tag{7}$$

then

$$F = S + jT \tag{8}$$

S_n^l is the outline feature of the nth image of the lth person, S is the outline feature of the mean image. They are got by adding up the weighted bit-plane 0,1,5,6, and 7. T_n^l is the texture feature of the nth image of the lth person, while T is the texture feature of the mean image. They are got by adding up the weighted bit-plane2,3, and 4. S^l is the class-mark of the lth person, it is got by computing the average of S_n^l. For the nth image of the lth person, we fuse its class-mark S^l with its texture feature T_n^l (see Eq. (6)), then get a new virtual face F_n^l, F_n^l is the new virtual face image. F is the new virtual mean face obtained by fusing S with T. j denotes the imaginary unit, that is $j^2=-1$. α_m, β_m

($\alpha_m \geq 0$, $\beta_m \geq 0$) are determined by trial and error. For different training samples, their values could be different.

For recognition stage, feature fusion is similar to training stage, the only difference is that since we don’t know to which class a test sample belongs, we can’t get its class-mark, thus we fuse its outline feature with its texture feature to form a virtual test face. Fig.4 gives an example of feature-fusion face.

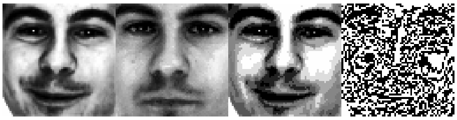


Fig. 4. A face image and its class-mark, outline feature and texture feature

In fig.4, from left to right, these 4 images are: the original face image x_n^l, class-mark S^l, the outline feature S_n^l and the texture feature T_n^l. Through Fig.4, it is seen that, class-mark eliminates the variations in facial expressions.

3.2 Generalized PCA

We define the feature-fusion space as $D = \{ \alpha + i\beta \mid \alpha, \beta \in R \}$. α and β are n-dimensional vectors in real space. Apparently, D-space is an n-dimensional complex-vector space. Now we define the inner product as $(x, y) = x^T y$, where $x, y \in D$. The complex space in which the above inner product has been defined is called unitary space. In unitary space, it is easy to verify that the total scatter S_t is an Hermite matrix, and it is nonnegative definite [4]. Thus, we can implement PCA in unitary space, and we call that Generalized PCA. In this paper, S_t can be rewritten as:

$$S_t = \frac{1}{LN} \sum_{l=1}^{L} \sum_{n=1}^{N} \left(F_n^l - F\right)\left(F_n^l - F\right)^T \tag{9}$$

Fig.5 shows the comparison between eigenfaces derived from feature-fusion faces (the first row) and eigenfaces derived from original faces(the second row). Since eigenfaces derived from the feature-fusion faces exist in complex space, we here show their modulus images. It is seen that, eigenfaces on the first row emphasize different local parts of a face, while eigenfaces on the second row reflect the holistic outline information of a face. Mainly because of that, our generalized PCA analysis on feature-fusion faces outperforms the conventional PCA, which is verified by the experiment in section 4.

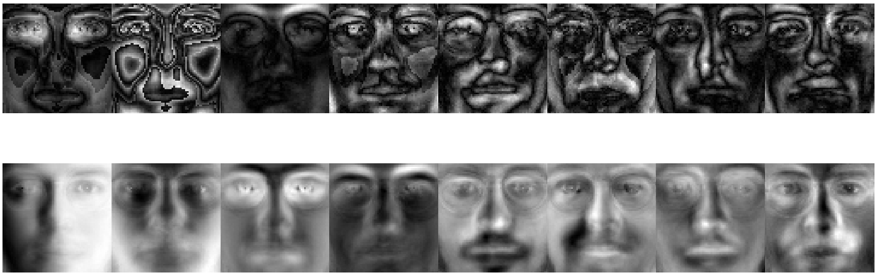


Fig. 5. Eigenfaces derived from feature-fusion faces and Eigenfaces derived from original faces

4 Experimental results

We combine SIC and BGPCA to form a joined face recognition algorithm and compare it with PCA [5] and Fisherface [6].

AR and Yale databases are used in the experiments. The AR database contains over 4,000 color face images of 126 people, including frontal views of faces with different facial expressions, lighting conditions and occlusions. We convert the color images into gray ones and only consider the full facial images. Yale database has 15 people, each has 11 images including different expressions, lighting conditions. Images in two databases are all normalized to 64×64 pixels.

4.1 Experiments on AR Database

In AR database, 10 images were chosen for each subject, for each individual, we chose his first 6 images to train and his last 4 images with left or right light on to test (see fig.6). To make it easy, every time we randomly chose 50 subjects to do the experiment, and repeated it for 3 times, then chose their average as the result.

The corresponding weights are: $\alpha_0=0$, $\alpha_1=0$, $\alpha_5=8.5$, $\alpha_6=5.8$, $\alpha_7=0.6$, $\beta_2=0$, $\beta_3=0$, $\beta_4=0.5$; $\alpha_0=0$, $\alpha_1=0$, $\alpha_5=7.8$, $\alpha_6=3.8$, $\alpha_7=1.3$, $\beta_2=0$, $\beta_3=0$, $\beta_4=1.4$; $\alpha_0=0$, $\alpha_1=0$, $\alpha_5=7.2$, $\alpha_6=6.3$, $\alpha_7=1.6$, $\beta_2=0$, $\beta_3=0$ and $\beta_4=0.2$. Generally, each weight is not confined to a certain value, but an optimal range. The results are listed in Table1.

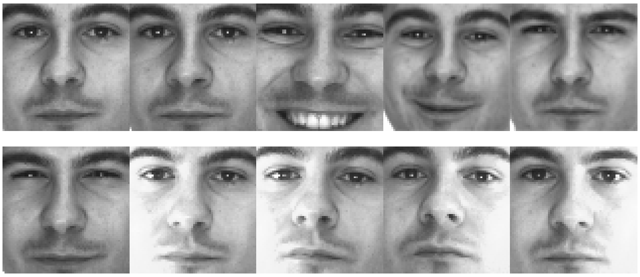


Fig. 6. Sample images of one subject in AR database

Table 1. Recognition rate on AR database

Method	Recognition rate (%)	Number of projection vectors
PCA	59.0	217
Fisher	68.5	76
Ours	81.5	41

In table 1, it is clear that PCA, Fisher face are sensitive to lighting. Our method can reduce the sensitivity with fewer projection vectors, but it has not reached an excellent result, in the future research, we need to do more work to improve it.

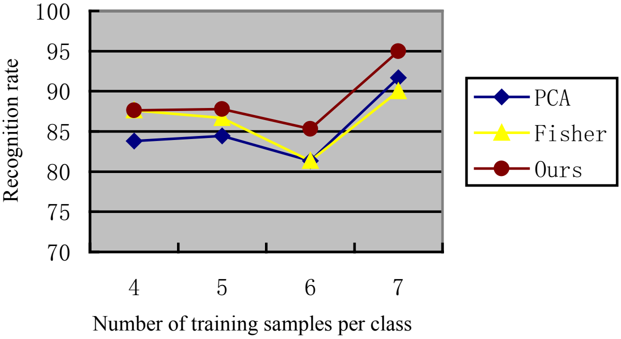


Fig. 7. Recognition rate on Yale database

4.2 Experiments on Yale Database

In Yale database, each individual has two images with one side lighting on, most of the images for one subject have center lighting. Considering the lighting condition of the left and right face can't be completely symmetry though with center lighting, so here we also use SIC on faces with center lighting. In this experiment, the number of training samples per class was set to 4,5,6 and 7 respectively, and the remaining images were used to test. The weights are respectively: α_0=1.46, α_1=3.78, α_5=0.05, α_6=0.54, α_7=0.45, β_2=0, β_3=0, β_4=0.23; α_0=0.41, α_1=1.78, α_5=0.13, α_6=5.33, α_7=1.62, β_2=0, β_3=0, β_4=0.19; α_0=1.25, α_1=0.21, α_5=0.1, α_6=3.82, α_7=1.11, β_2=0, β_3=0, β_4=0.22; α_0=3.16, α_1=0.31, α_5=0.11, α_6=1.42, α_7=2.14, β_2=0, β_3=0 and β_4=0.78; The corresponding recognition result is illustrated in Fig.7.

5 Conclusions

In this paper, we proposed two algorithms. One is an image-preprocessing algorithm that is easy to be implemented, i.e. Symmetrical Image Correction (SIC); the other is an extended PCA algorithm, i.e. Generalized PCA based on bit-plane feature fusion. We combine these two algorithms to perform face recognition. The proposed recognition method is effective for realizing robust face recognition under one side lighting conditions.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant no.60172022 and the Outstanding Young and Middle Age Scientist Fundation of Shandong under no 03BS001.

References

1. Moses Y, Adini Y, and Ullman S. "Face recognition:The problem of compensating for changes in illumination," IEEE Transactions on PAMI,1997,19(7):721-732.
2. Lemieux, A., Parizeau, M.,. Experiments on eigenfaces' robustness. Proceedings of The 16th International conference on Pattern Recognition. 2002, pp.421-424.
3. Shakunaga, T., Sakaue, F., 2002. Natural image correction by iterative projections to eigenspace constructed in normalized image space. In: The 16th International conference on Pattern Recognition. pp.648-651.
4. Yang J., and Yang J.Y. "Generalized K-L transform based combined feature extraction," Pattern Recognition, 2002,35(1): 295-297.
5. Turk M A, and Pentland A P. "Eigenfaces for recognition," J Cognitive Neurosci, 1994, 3(1): 71-86.
6. Belhumeur P N,Hespanha J P,Kriegman D J. "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," IEEE Transactions on PAMI,1997,19(7):711-720.

E-2DLDA: A New Matrix-Based Image Representation Method for Face Recognition*

Long Fei[1], Dong Huailin[1], Fan Ling[2], and Chen Haishan[1]

[1] Software School, Xiamen University, Xiamen 361005, China
{flong, hldong, hschen}@xmu.edu.cn
[2] Tan Kah Kee College, Xiamen University, Zhangzhou 363105, China
fling@xujc.com

Abstract. Two-dimensional linear discriminant analysis (2DLDA) was recently developed for face image representation and recognition by adopting the idea of image projection in 2DPCA. 2DLDA outperforms traditional LDA mainly in terms of feature extraction speed. Unfortunately, 2DLDA needs to use large numbers of features to represent an image sample, causing storage requirements are heavy and also feature matching process is time-consuming. Against this problem, we discuss in this paper a new image representation scheme called Enhanced 2DLDA (E-2DLDA) for face recognition. The main strategy adopted in our method is that two image projections are applied to an image sample jointly, so the dimensions of extracted feature matrix along both horizontal direction and vertical direction get compressed, and finally the total number of features can be reduced to a great extent. The experimental results on ORL database show that this method remarkably outperforms existing 2DLDA in terms of speed of feature matching and storage requirements of features.

1 Introduction

Automatic face recognition (AFR) [1] has been a very hot research area of computer vision, pattern recognition and machine learning, especially for past about 10 years. The current state-of-the-art face recognition can be characterized by a family of subspace approaches, such as PCA, LDA and ICA [2][3] etc. in which, eigenfaces (PCA) [4] and fisherfaces (PCA+LDA) [5] have been used as two famous baselines for evaluating other algorithms. Traditional subspace methods are all vector-based, which means that when we apply it to image recognition problem, an image must be firstly transformed into a high-dimensional vector by concatenating all rows or columns of the image. The resulting image vectors usually lead to a high dimensional image vector space, in such a case, feature extraction will be a very difficult task due to so called curse-of-dimensionality and small sample size (SSS) problem encountered.

Recently, an image projection technique termed two-dimensional principal component analysis (2DPCA) [6] was proposed, which treats images as 2D matrices

[*] This work was partially supported by the Program of 985 Innovation Engineering on Information in Xiamen University (2004-2007).

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 236 – 243, 2006.

rather than 1D vectors (matrix-based), as a result, the speed of image feature extraction is improved significantly compared with conventional PCA. Motivated by the image projection idea in 2DPCA, 2DLDA [7] method was naturally developed later, which replaces total scatter criterion used in 2DPCA with Fisher linear projection criterion to find out more discriminating projections, so 2DLDA can usually produce a better recognition accuracy than 2DPCA. However, both 2DPCA and 2DLDA need to use large numbers of features to represent an image, causing storage requirements are heavy and also feature matching process is time-consuming.

In this paper, we discuss a new image representation scheme called Enhanced 2DLDA (E-2DLDA) for face recognition. The main strategy adopted in our method is that seeking two sets of optimal projection vectors through respectively regarding rows and columns of image as objects for analysis, and forming two linear transforms (image projections), which are applied to an image sample jointly, as a result, the dimensions of extracted feature matrix along both horizontal direction and vertical direction get compressed, and finally the total number of features can be reduced to a great extent. In addition, because not only row-objects but also column-objects in images get analyzed during two projections, more discriminating information can be exploited in feature extraction, which leads to an enhanced recognition accuracy eventually. The effectiveness of our method is demonstrated on the ORL database of faces. The rest of this paper is organized as follows, in Section 2 we briefly review the existing method of 2DLDA, in Section 3 we present the motivation of our E-2DLDA method, and also the E-2DLDA based face recognition algorithm, in Section 4, we give experimental results of face recognition on ORL database, and a conclusion is given in Section 5.

2 Review on 2DLDA

Let **I** denotes an m×n image, and **u** denotes an n-dimensional column vector, called projection vector or projection axis. The idea of image projection [6] can be simply described as: projecting **I** onto **u** by the following linear transformation,

$$\mathbf{y} = \mathbf{I}\mathbf{u} \tag{1}$$

the m-dimensional projected vector **y** in Eq.1 is called the projected feature vector of image **I**.

Only one projection vector is not enough for the goal of feature extraction, usually, we need to select a set of projection vectors $\mathbf{u}_1$, $\mathbf{u}_2$,…, $\mathbf{u}_d$. 2DLDA seeks such vectors with the classic Fisher linear projection criterion, that is to say, the optimization objective is that the ratio of between-class scatter to within-class scatter of projected samples is maximized. Consider the training set $\{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_L\}$, where $\mathbf{I}_i \in R^{m\times n}$ denotes an image matrix, belonging to one of C classes $D_1, D_2, \dots, D_C$. 2DLDA defines within-class scatter matrix $\mathbf{G}_W$ and between-class scatter matrix $\mathbf{G}_B$ based directly on image matrices as bellow,

$$\mathbf{G}_W = \sum_{i=1}^{C} \sum_{\mathbf{I}\in D_i} (\mathbf{I} - \bar{\mathbf{I}}_i)^T (\mathbf{I} - \bar{\mathbf{I}}_i) \tag{2}$$

$$\mathbf{G}_B = \sum_{i=1}^{C} (\bar{\mathbf{I}}_i - \bar{\mathbf{I}})^T (\bar{\mathbf{I}}_i - \bar{\mathbf{I}}) \tag{3}$$

where, $\bar{\mathbf{I}}_i$ is the mean of class *i*, $\bar{\mathbf{I}}$ is the mean of all training samples, *C* is the class number. Thus, the objective function of 2DLDA can be written as,

$$J(\mathbf{u}) = \frac{\mathbf{u}^T \mathbf{G}_B \mathbf{u}}{\mathbf{u}^T \mathbf{G}_W \mathbf{u}} \tag{4}$$

So, the optimal projection vectors should chosen as the first *d* eigenvectors $\mathbf{u}_1$, $\mathbf{u}_2$, … , $\mathbf{u}_d$ in the following generalized eigenvalue problem,

$$\mathbf{G}_B \mathbf{u} = \lambda \mathbf{G}_W \mathbf{u} \tag{5}$$

Project **I** onto these vectors respectively, $\mathbf{y}^{(k)} = \mathbf{I}\mathbf{u}_k, k = 1,2,\cdots,d$, the projected feature vectors can compose a matrix $\mathbf{Y} = [\mathbf{y}^{(1)} \cdots \mathbf{y}^{(d)}] \in R^{m \times d}$, which is called the extracted feature matrix from **I**. According its definition, we know that $\mathbf{G}_W$ is a nonsingular matrix, so 2DLDA method has successfully overcome the problem of singularity of within-class scatter matrix encountered in fisherfaces [5] method. In addition, the size of two scatter matrices in Eq.5 is not large in general, so the training time for feature extraction will also be much less than traditional vector-based methods.

3 E-2DLDA

3.1 Motivation of E-2DLDA Method

According to the image projection idea adopted by 2DLDA, projecting an *m*×*n* image matrix **I** onto *d* *n*-dimensional column vectors $\mathbf{u}_1$, $\mathbf{u}_2$, … , $\mathbf{u}_d$ will produce an *m*×*d* feature matrix **Y**. Where, *m* is the row number of an image matrix, so even if *d* is small, the total feature number *md* is quite large. For example, to an 112×92 image, if we choose first 5 optimal vectors to perform image projection, the number of extracted features will also achieve 560(=112×5).

Comparing the size of the extracted feature matrix **Y** to that of the original image sample **I**, we find that the current image projection technique like 2DLDA can only compress the dimensionality of image along horizontal direction, thus causing an incomplete dimensionality reduction from the viewpoint of feature extraction. The reason behind this drawback can be further explained. From the definitions of within-class scatter matrix $\mathbf{G}_W$ in Eq.2 and between-class scatter matrix $\mathbf{G}_B$ in Eq.3, we know that the size of $\mathbf{G}_W$ and $\mathbf{G}_B$ is both *n*×*n* (n is the dimensionality of rows in an image), which means that the second order statistics contained in these two scatter matrices are actually exploited from the "row samples" of images. Therefore, 2DLDA method implicitly regards only rows of image as objects for analysis, as illustrated in Fig.1(a). In other words, this method just partially exploits useful statistical information contained in image data.

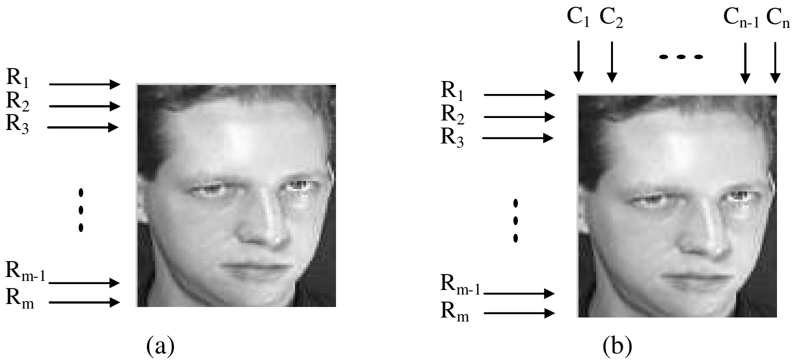


Fig. 1. Two implicit analysis manners in 2DLDA(a) and E-2DLDA(b)

Based on above considerations, here, we discuss a new scheme called Enhanced 2DLDA (E-2DLDA). As illustrated in Fig.1(b), firstly, all rows of image samples are regarded as samples, and based on which a set of projection vectors $\mathbf{P} = [\mathbf{p}_1\ \mathbf{p}_2\ \cdots \mathbf{p}_d] \in R^{n\times d}$ are learned, this step is just the same to that of 2DLDA. Secondly, columns are further regarded as objects to analyze, then another set of projection vectors $\mathbf{Q} = [\mathbf{q}_1\ \mathbf{q}_2\ \cdots \mathbf{q}_k] \in R^{m\times k}$ are learned. Lastly, the feature extraction is done by a manner of joint image projection as below,

$$\mathbf{Y} = \mathbf{Q}^T \mathbf{IP} \tag{6}$$

the transformed result is $\mathbf{Y} \in R^{k\times d}$, so the number of features is *kd*. Because *k* and *d* are much smaller than row number and column number of images, this joint projection can produce an effective dimensionality reduction. In addition, the set of projection vectors $\mathbf{q}_1$, $\mathbf{q}_2$, … , $\mathbf{q}_k$ provide more discriminatory information for feature extraction compared with existing 2DLDA, so the effectiveness of extracted features is also enhanced. The detailed face recognition algorithm based on the proposed E-2DLDA method will be presented in below.

3.2 E-2DLDA Based Face Recognition Algorithm

In E-2DLDA algorithm, a two-step strategy is used to compute the first transform matrix **P** and the second transform matrix **Q**, but it doesn't matter to change the computation order of them. Based on these two transform matrices, face image representation (or image feature extraction) will be carried out with a manner of joint image projection, and then followed by feature matching and classification. So, the E-2DLDA based face recognition algorithm can be decomposed into 4 steps in all, which is described as bellow.

Step 1. Computation of $\mathbf{P}$ --- For a given training face samples $\{\mathbf{I}_1, \mathbf{I}_2, \cdots, \mathbf{I}_L\}$, compute within-class scatter matrix and between-class scatter matrix according to Eq.2 and Eq.3 respectively, here denoted as $\mathbf{G}_W^{\mathbf{P}}$ and $\mathbf{G}_B^{\mathbf{P}}$, solve the generalized eigenvalue problem $\mathbf{G}_B^{\mathbf{P}}\mathbf{u} = \lambda\mathbf{G}_W^{\mathbf{P}}\mathbf{u}$, and choose the first d eigenvectors to compose the transform matrix $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_d]$.

Step 2. Computation of **Q** --- Compute within-class scatter matrix $\mathbf{G}_W^{\mathbf{Q}}$ and between-class scatter matrix $\mathbf{G}_B^{\mathbf{Q}}$ as below,

$$\mathbf{G}_W^{\mathbf{Q}} = \sum_{i=1}^{C}\sum_{\mathbf{I}\in D_i}(\mathbf{I}-\bar{\mathbf{I}}_i)(\mathbf{I}-\bar{\mathbf{I}}_i)^T \tag{7}$$

$$\mathbf{G}_B^{\mathbf{Q}} = \sum_{i=1}^{C}(\bar{\mathbf{I}}_i-\bar{\mathbf{I}})(\bar{\mathbf{I}}_i-\bar{\mathbf{I}})^T \tag{8}$$

Note that the computation methods of within-class scatter matrix and between-class scatter matrix in Eq.7 and Eq.8 are subtly different from those in Eq.2 and Eq.3. Actually, we just replace the original training image samples with their transpose matrix versions, and just this simple change let columns of images be analyzed, so dimensions of image along vertical direction can be reduced further, and also additional statistical information can be exploited. Solve the generalized eigenvalue problem $\mathbf{G}_B^{\mathbf{Q}}\mathbf{u} = \lambda\mathbf{G}_W^{\mathbf{Q}}\mathbf{u}$, and choose the first k eigenvectors to compose another transform matrix $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k]$.

Step 3. Feature extraction --- The transform matrices **P** and **Q** (more exactly, two sets of optimal vectors) computed above are used for performing feature extraction. For a given image sample **I**, through following joint image projection

$$\mathbf{Y} = \mathbf{Q}^T\mathbf{I}\mathbf{P} = [\mathbf{q}_1\ \mathbf{q}_2 \cdots \mathbf{q}_k]^T\mathbf{I}[\mathbf{p}_1\ \mathbf{p}_2 \cdots \mathbf{p}_d] \tag{9}$$

we get a k×d small scale matrix **Y**, which is called the feature matrix of image **I**.

Step 4. Feature matching and classification--- After joint image projection based on E-2DLDA, a feature matrix is extracted from each image. In recognition stage, nearest neighbor based matching is used to classify unknown samples to one of C given classes $D_1, D_2, \dots, D_C$. The similarity between two images is measured by Euclidean distance,

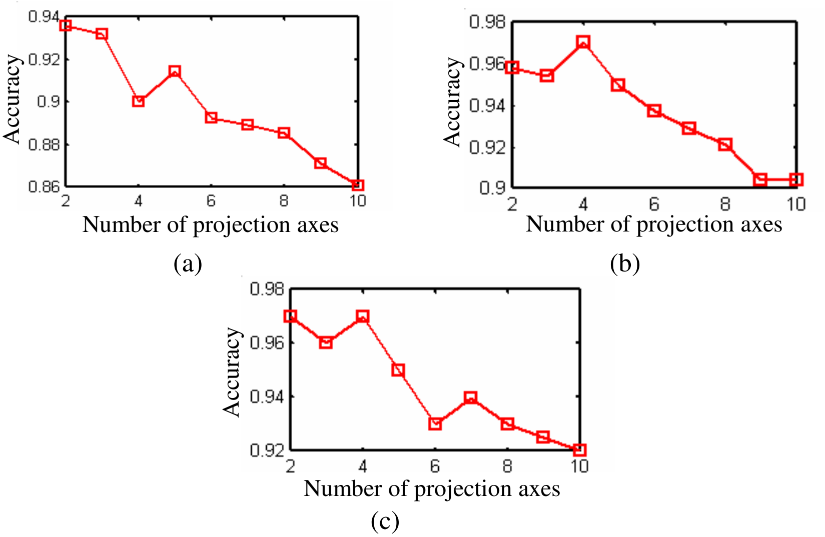
$$d(\mathbf{Y}_1,\mathbf{Y}_2) = (\sum_{i=1}^{k}\sum_{j=1}^{d}[\mathbf{Y}_1(i,j)-\mathbf{Y}_2(i,j)]^2)^{1/2} \tag{10}$$

Assume there is a gallery set including M feature matrices $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_M$, each of them is assigned a given class D_i. Suppose an unknown sample with feature matrix **A**, if $d(\mathbf{A},\mathbf{A}_l) = \min_j d(\mathbf{A},\mathbf{A}_j)$, and $\mathbf{A}_l \in D_k$, then the resulting decision is $\mathbf{A} \in D_k$.

4 Experimental Results

We use the ORL database of faces (http://www.cam-orl.co.uk) to comparatively evaluate the performance of the proposed E-2DLDA method and the existing 2DLDA method [7]. The ORL database contains 40 persons with 10 images (112×92) per person. The images are taken at different times, with varying lighting conditions, facial expressions and facial details (glasses/no glasses). All persons are in the upright, frontal position, with tolerance for some side movement.

The face recognition test is designed as follows. The whole ORL database is partitioned into two complementary parts, i.e., training set and testing set, and the training set is also used as a gallery set in nearest neighbor based matching. For this purpose, we randomly select N samples from each person for training, and the rest for testing. In the test, recognition performance by using different amount of training samples is taken into account. More specifically, we use $N = 3$, $N = 4$ and $N = 5$ samples from each person to train a set of optimal projection vectors (axes) for feature extraction. In the three partitions, the numbers of training samples are respectively 120, 160 and 200, and the numbers of testing samples are therefore 280, 240 and 200 respectively. The performance yielded by existing 2DLDA method and our E-2DLDA method is illustrated in Fig.2 and Fig.3 respectively. Fig.2 plots the recognition accuracy with increasing projection axes used in image projection, where $d = 2,3,\ldots,10$. Fig.3 plots the recognition accuracy with different pairs of (d, k) used in joint image projection, and $d = 2,3,\ldots,10$, $k = 2,3,\ldots,10$, where d is the number of projection axes in transform **P**, and k is the number of projection axes in transform **Q**.



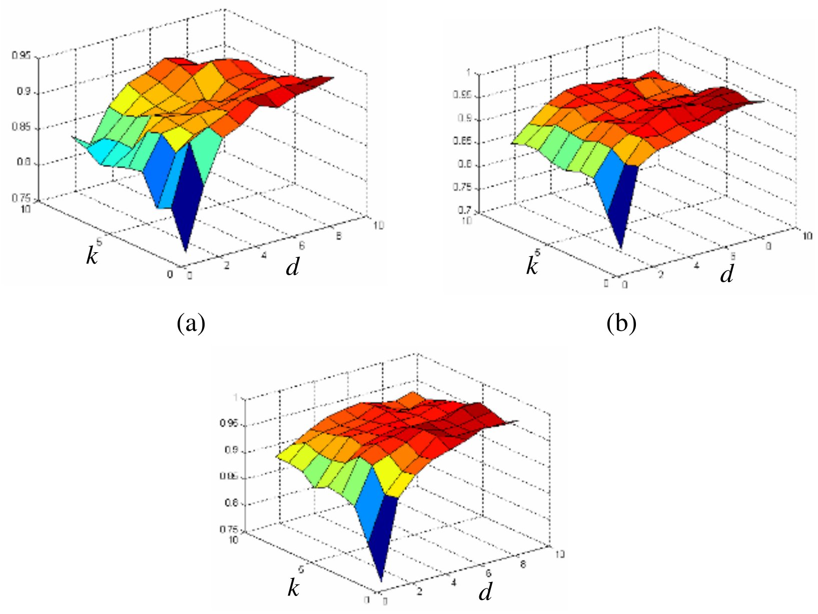
(a)

(b)

(c)

Fig. 2. Recognition performance of 2DLDA method on ORL database. (a), (b) and (c) are the test results for N=3 N=4 and N=5 respectively.

Table 1 presents the top correct recognition rates of 2DLDA and E-2DLDA, as well as the number of projection axes, the number of extracted features, CPU time for training (optimal projection vectors) and CPU time for feature matching when the top recognition accuracy appears. As observed in Table.1, for each partition our E-2DLDA outperforms existing 2DLDA, especially, the advantage is significant in terms of CPU time for feature matching and the amount of features used in recognition. In addition,



(a) (b)

(c)

Fig. 3. Recognition performance of E-2DLDA method on ORL database. (a), (b) and (c) are the test results for *N*=3 *N*=4 and *N*=5 respectively.

Table 1. Comparative top recognition accuracy, number of projection axes, number of features, CPU time for training (optimal projection vectors) and feature matching of 2DLDA and E-2DLDA(CPU: Pentium III 1.2GHz, RAM: 256MB)

Methods	Training set/testing set partitions	Number of projection axes	Number of features	Recognition accuracy	CPU time for training	CPU time for feature matching
2DLDA	*N* = 3	d = 2	112×2	93.6%	0.7	1.1
E-2DLDA		d = 2 k = 10	2×10	94.3%	1.8	0.1
2DLDA	*N* = 4	d = 4	112×4	95.8%	0.8	2.5
E-2DLDA		d = 2 k = 9	2×9	97.5%	2.2	0.1
2DLDA	*N* = 5	d = 4	112×4	97.0%	1.0	2.7
E-2DLDA		d = 2 k = 8	2×8	99.0%	2.6	0.1

because row-objects and column-objects in images are analyzed respectively, more discriminating information is exploited in feature extraction procedure, therefore yielding a better recognition accuracy also. Meanwhile, we also observe that E-2DLDA is not as efficient as 2DLDA in terms of CPU time for training, the reason is that two sets of optimal projection vectors need to be trained for feature extraction in E-2DLDA, but there is only one set to be trained in 2DLDA. Fortunately, the training speed actually has nothing to do with the recognition speed when the system work in real application situations, because once the training is finished, feature extraction later is simply a linear transform during the period of recognizing a probe face image.

5 Conclusion

An E-2DLDA matrix-based image representation method for face recognition is proposed in this paper. The key strategy adopted in E-2DLDA method is seeking two sets of optimal projection vectors by respectively regarding rows and columns of image as objects for analysis, and then jointly applying these two linear transforms to an image sample to get a compact as well as discriminatory representation. Experimental results on ORL database show that the proposed method has significant advantage over existing 2DLDA in terms of the speed of feature matching and also the storage requirements of features.

References

1. Zhao, W., Chellappa, R., Phillips, P.: Face Recognition: A Literature Survey. ACM Computing Surveys, vol. 35. (2003) 399-458.
2. Bartlett, M.S., Movellan, J.R., et al.: Face recognition by independent component analysis. IEEE Trans. On Neural Networks, vol. 13. (2002) 1450-1464.
3. Long Fei, He Jinsong, Ye Xueyi, et. al.: Discriminant Independent Component Analysis as a Subspace Representation. Journal of Electronics(China), vol. 23. 2006 103-106.
4. Turk, M., et. al.: Eigenfaces for face recognition. Journal of Cognitive Neuroscience, vol. 3. (1991) 71-86.
5. Belhumeour, P.N., Hespanha, J.P., Kriegman, D.K.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Trans. On PAMI, vol. 19. (1997) 711-720.
6. Yang, J., et al.: Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition. IEEE Trans On PAMI, vol. 26. (2004) 131-137.
7. Ming Li, Baozong Yuan.: 2D-LDA: A novel statistical linear discriminant analysis for image matrix. Pattern Recognition Letters, vol. 26. (2005) 527-532.

Adaptive Color Space Switching Based Approach for Face Tracking

Chuan-Yu Chang[1], Yung-Chin Tu[2], and Hong-Hao Chang[1]

[1] Department of Electronic Engineering,
National Yunlin University of Science & Technology, Douliou, Yunlin, Taiwan
[2] Department of Electronic Engineering,
Kun Shan University, Yung Kang, Tainan, Taiwan
chuanyu@yuntech.edu.tw

Abstract. In this paper, a support vector machine (SVM) based adaptive color switching for human face tracking is proposed. The color space is switching to the most appropriate color space model (CSM) according to circumstance conditions adaptively. Recently, many face tracking algorithms used empirical skin color model to discriminate skin/non-skin regions. These skin color models not consider illumination variation and result in less capacity to model skin color distribution. In this work, four color spaces and Laws texture extracted from face image database are used to train each SVM independently. In the preprocessing, the discrete wavelet transform (DWT) refines the face features would concentrate important features and reduce the computational complexity. Then, the features are transformed into four CSMs for SVMs which provide good generalization through optimal hyperplane. In testing, we perform quality measurement method to evaluate the face tracking performance and aggregating each SVM classification results to color space switching. Experimental results show that the proposed method would switch to the most appropriate color space according to quality measurement, automatically.

Keywords: Adaptive Color Space Switching, Face Tracking, Support Vector Machine.

1 Introduction

Recently, many researchers have been investigated efficient and powerful algorithms for moving object tracking. Many literatures discussed CSM based on computer vision and neural network for face detection [7, 13] and recognition [5, 10, 15]. For face detection, there are various methods included facial features and skin color based algorithms [14]. Intuitively, color is an important feature of human face. Using skin color features to track faces reveal several advantages [16]. Especially, skin color is size and orientation invariant for human face under stable illumination conditions. Unfortunately, skin color is sensitive to illumination form human visual perception. On the other hand, human face tracking using skin color feature encounters several problems such that representation of a face obtained from a camera is influenced by

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 244 – 252, 2006.

many factors and each human face has different properties of skin color distribution. Stern [9] is first attempt to adaptively select CSMs throughout a tracking sequence. He used the back projection and flesh probability image to track the skin color. The method needs specific skin region manually before tracking. Further, the skin distribution of pre-selection region not presents the human skin color distribution at all even leads to tracking loss.

The SVM is an optimal classification and regression technique proposed by Vapnik and his group at AT&T Bell Laboratories [2]. The SVM learns a separating hyperplane to maximize the margin between training set and to provide good generalization performance. Nowadays, it has been successfully applied to many fields such as the object recognition [10, 17], pattern classification [4], regress in estimation [1], and environment illumination learning [16].

The idea of the SVM ensemble has been proposed in Ref. [18]. They used the boosting technique to train each individual SVM and took trained SVMs to build SVM committee. The SVM committee based on the bagging and boosting techniques to improve performance has discussed in [8]. The mixture of expert models and methods of constructing committee machine are reviewed in [19]. The SVM ensemble partition the whole training observation into several subsets and to make each individual trained SVM on their respective training subsets [6]. We propose SVM committee to color switching for tracking. Each SVM trained by specific observation is transformed to different color spaces. The method makes trained examples view as many aspects and to improve results by quality measurement. We expect that the method can improve the classification performance and reduce the computational complexity.

This paper is organized as follows. Section 2 presents the color space switching algorithm and related theorems includes system architecture, discrete wavelet transform, color space selection and feature extraction, basic idea of SVM and quality measurement. Then, section 3 provides experimental results and section 4 makes some conclusions.

2 Color Space Switching Algorithm

Many researchers used CSMs for object tracking and recognition based on color distribution of interesting object. The main objective of this paper is to improve the face tracking performance through quality measurement. In multiple color spaces, it is necessary to reduce dimension of color spaces to concentrate important features. For instance, principal component analysis (PCA) and vector quantization (VQ) are popular for dimensional reduction. However, these dimensional reduction methods were time consuming caused the tracking algorithm inefficiency. Therefore, in this paper, we attempt to develop an automatic CSM switching system and propose an efficient quality measurement algorithm for face tracking.

2.1 System Architecture

SVM committee is a collection of several classifiers whose decisions are combined into some ways to classify the examples. Sometime, the support vectors obtained from the training data is less sufficient to classify unknown samples correctly. Thus, it is not guarantee that single SVM provides the global optimal classification and good generalization over all test examples. It is possible to lead faulty tracking performance. In order to overcome these drawbacks, we propose a SVM committee which integrated each decision to select the most appropriate CSM. Figure 1 shows the architecture of SVM committee which consists of DWT, SVM classifiers and quality measurement. In this work, we use Haar basis function which provides symmetric and orthogonal attributes to analyze image multi-resolution. The LL sub-band achieves dimensional reduction and noise decompression, while these high-frequency subbands are noisy and messy. Therefore, the LL sub-band is selected for further processing. During the training phase, each SVM is trained independently in different color spaces and texture features. All SVM results would aggregate results to select the most appropriate color space.

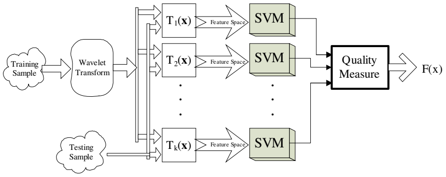


Fig. 1. The architecture of SVM committee for color switching

2.2 Color Space Selection and Feature Extraction

In our system, training sample $\mathbf{x}(n)$ is obtained from face images in RGB space. Through the color transform function $T(\mathbf{x})$, the RGB coordinate is converted into some color coordinates such as YCbCr, normalized RGB, XYZ and YIQ. Such color spaces are highly correlation each other with coordinate rotation and linear transformation from RGB space. Among them, YCbCr and YIQ isolate the illumination apart from chrominance. In order to reveal interested properties of object, we convert face image into different coordinates for face tracking under different environments.

The texture energy measures developed by Kenneth Ivan Laws have been used for diverse applications [11]. These measures are computed by applying small convolution kernels to a digital image, and then performing a nonlinear windowing operation. In this paper, L5E5, R5E5 and W5W5 are selected as features through observation and analysis.

2.3 Support Vector Machine Classification

Consider the training set $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$ and testing set $\{(\mathbf{x}_i, d_i)\}_{i=N+1}^{N+M}$, where N is the size of training data, M is size of testing data, $\mathbf{x}_i$ is the input space for *i*th example and $d_i \in \{+1,-1\}$ is the corresponding desired responses. The basic idea of SVM classification is to find an optimal separating hyperplane that maximizes the margin between two classes. The margin is defined as the distance of the closet point to the separating hyperplane and input space $\mathbf{x}$ is mapped into a high dimensional feature space through kernel function $\Phi(\mathbf{w})$. In the case of separable binary classification problem, the discriminant function of optimal hyperplane represents a multi-dimensional decision plane in the input space. The discriminant function is defined by following

$$g(\mathbf{x}) = \mathbf{w}_o^T\mathbf{x} + b_o \tag{1}$$

where $\mathbf{w}_o$ and b_o denotes the optimal values of the weight vector and bias, respectively. In SVM, the optimal separating hyperplane is determined by support vectors $\mathbf{x}^{(s)}$ that lie closest to the decision surface. The optimal hyperplane is required to satisfy the following constraints

$$\begin{cases} d_i(\mathbf{w}^T\mathbf{x}_i + b) \ge 1 & \text{for } i = 1,2,...,N \\ \mathbf{\Phi}(\mathbf{w}) = \dfrac{1}{2}\mathbf{w}^T\mathbf{w} \end{cases} \tag{2}$$

In order to solve Eq. (2), we need to construct a set of functions and implement the classical risk minimization on the set of function. Here, a Lagrangian method is used to solve such optimization problem with constraints. The Lagrangian function is defined as follows:

$$J(\mathbf{w},b,\alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N}\alpha_i\left[d_i(\mathbf{w}^T\mathbf{x}_i + b) - 1\right] \tag{3}$$

where the auxiliary variable α_i is called Lagrange multiplier and its value is positive. The solution of the constrained optimization problem is determined by the saddle point of the Lagrangian function. Then, we may reformulate the objective function $J(\mathbf{w}, b, \alpha)$ of the optimal problem to dual form defined in Eq. (4):

$$Q(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j \mathbf{x}_i^T\mathbf{x}_j \tag{4}$$

The kernel function plays an important role to the approximation of nonlinear mapping and externs SVM to handle nonlinear separating hyperplane. In this work, the Gaussian kernel function $\mathrm{K}(\mathbf{x}, \mathbf{x})$ is adopted to map the data space into high dimension feature space. The LIBSVM software is available in [3]. The Gaussian kernel function is defined as

$$\mathrm{K}(\mathbf{x}, \mathbf{x}^{(s)}) = \exp(-\left\|\mathbf{x} - \mathbf{x}^{(s)}\right\|^2 / 2\sigma^2) \tag{5}$$

3 Color Space Quality Measurement

Since the quality of the segmented human face is varying under different CSM. To select the most appropriate CSM, a quality measure is proposed. In this work, the interesting regions have two parts: (i) W_{in}, the area of internal rectangular window containing the face, and (ii) W_{out} , the area of external rectangular window which exclude W_{in} region illustrated in Fig. 2. Let the coordinates of the common center (c_x, c_y) is the location of central pixel in W_{in} and W_{out} . The rapid face detection scheme which using a set of rotated haar-like features was proposed by [12]. The method can be calculated very efficiently and lower average false alarm rate. A novel post optimization procedure for a given boosted cascade also used to improving accuracy and performance. The quality measurement of face tracking result Q_k is defined in Eq. (6).

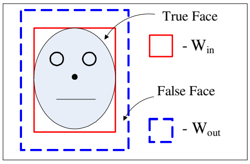


Fig. 2. Internal and external window for quality measurement

$$Q_k = \frac{p_k^{in}}{p_k^{out}} = \frac{c_k^{in} / W_{in}}{(c_k^{in} / W_{in}) + (c_k^{out} / W_{out})} \tag{6}$$

Let k represent the k_{th} color space. The W_{in} and W_{out} denote the area of internal and external rectangular windows, respectively. The c_k^{in} and c_k^{out} is the detected face pixel number of the internal and external rectangle window, respectively. So, the p_k^{in} represents the probability of face pixel of k_{th} color space located in the internal window and p_k^{out} is the probability of external window of face. The higher Q_k value is the higher accuracy of the segmented face.

4 Experimental Results

In our experiments, all of face images are obtained from the Psychological Image Collection at Stirling (PICS) image database. The training face images were segmented apart from background manually. The real world scenes contain one or more human with frontal face in tracking sequences are used for testing. First, we perform face detection algorithm to locate face region. The internal rectangle is the detected face region and external region are augmented the 30% of width and height of internal rectangle. We assure the classification accuracy of each SVM provide upper 99.9% in training data to offer high accuracy.

4.1 Feature Extraction

The Laws texture energy measurement determined by the property of specific kernels which assess average gray-level, edges, spots, ripples, and waves. These selected kernels were convoluted with face image to find the better kernels which can extract apparent face from background. After statistics, we analyze the fluctuation of face features by the first derivative of energy, $\partial f / \partial L$, $\partial f / \partial E$, $\partial f / \partial S$, $\partial f / \partial R$ and $\partial f / \partial W$ where f is the Laws texture energy. The selected five texture kernels are sifted from twenty-five texture kernels which have average, edge, spot, ripple and wave properties.

4.2 Tracking Sequence Results and Quality measure

We perform face tracking algorithm to locate face regions and calculating the quality measurement in each CSMs. Our experimental environment is indoors with clutter objects. There are two light sources in this experiment, a moveable and tunable table-lamp is used to simulate light variation, and a fixed fluorescent lamp located on the ceiling is used to provide stationary illumination. The proposed method would select the most appropriate results and switch to corresponding color space automatically according to quality measure method.

In the case , the light of desk-lamp spots on the face partially. Figure 3(a) is the original image; Figure 3(b-c) are the segmentation results of YCbCr, YIQ, RG and XY color space, respectively. Observing the tracking result of RG space in Fig. 3(d), the RG color space is affected by both light sources greatly and others color spaces resulted in better solutions with slight difference. These results illustrate the face tracking under the RG color space with weak ability to against light variation. The average, maximum and minimum sensitivity $\hat{Q}_k$ in different color spaces is shown in Table 1.

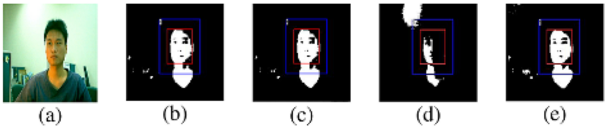


Fig. 3. Case 1:the tracking result with hard brightness

Table 1. The sensitivity of tracking sequence from different color spaces

	CbCr	IQ	RG	XYZ
Average $\hat{Q}_k$	0.7211	0.6995	0.4487	0.6643
Max $\hat{Q}_k$	0.7666	0.7459	0.5548	0.7482
Min $\hat{Q}_k$	0.6420	0.6019	0.3047	0.6167

Table 2. The performance of face tracking from different color spaces

	CbCr	IQ	RG	XYZ
True Positive	0.4451	0.4367	0.0468	0.4781
False Positive	0.0671	0.0648	0.0803	0.0777

Figure 4 shows the tracking sequence under CbCr space. The illumination is varied to change the brightness of face region. Figures 4(a-f) are the image sequence which obtained from web camera, Figs. 4(g-l) are the corresponding segmentation results. Table 2 shows the average true positive and false positive rate for each color spaces. The performance of CbCr, IQ and XYZ space are similar to each others but the RG space is still not a good solution. Figure 4 shows the quality measurement of four color spaces in a real world tracking sequence. The CbCr space reveals better performance at the first 35 frames and the last 15 frames, while XYZ space reveals better performance from 36th to 45th frames. In this case, a man moves his position at the 35th frame causes the environmental illumination varied. In addition, the light of desk-lamp turns to soft brightness at the 40th frame. In order to achieve real-time tracking, the color space switching mechanism is triggered for every ten frames. According to the quality measurement, the proposed algorithm selects and switch to the most appropriate color space automatically. Thus, the CbCr space is switched to XYZ color space at the 40th frame and switched back to CbCr space at the 50th frame. The comparison of with/without adaptive color space switching is shown in Table 3. The experiment result shows the performance of adaptive color space switching is better than popular CbCr color space shown in Table 3.

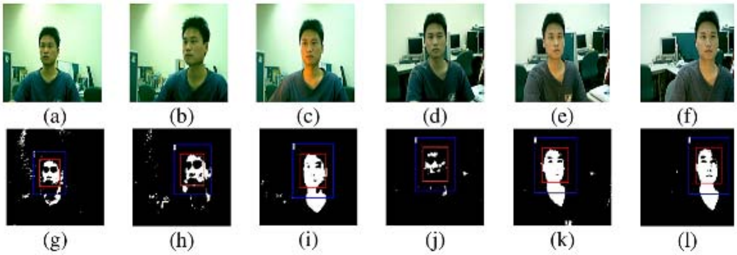


Fig. 4. The tracking sequence using CbCr space

Table 3. Quality evaluation with/without adaptive colors space switching

	Average Quality	Average Sensitivity	Average Specificity
with-ACSS	0.871000	0.827602	0.893607
CbCr without-ACSS	0.764214	0.756352	0.793305

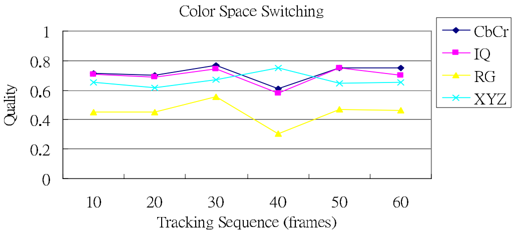


Fig. 5. The quality of tracking sequence under four color spaces

5 Conclusions

In this work, we proposed a SVM-committee-based color space switching algorithm for face tracking. In this learning machine, CSMs of skin and Laws texture energy of faces are selected for training SVMs. The SVM-committee integrates multi-classifier with high accuracy and generalization. In addition, we define a quality measurement to select the most appropriate CSM for face tracking. The experimental results shown the proposed method was validated under different object behaviors and environmental variation such as camera motion, background change, object motion and brightness variation. The experimental results also concluded that RG color space is highly depend on illumination.

References

1. Smola, A.J. and Scholkopf, B.: A Tutorial on Support Vector Regression. NeuroCOLT Technical Report NC-Tr-1998-030, Royal Holloway College, University of London, UK, (1998)
2. Cortes, C. and Vapnik, V.: Support-Vector Network, vol.20. Mach. Learn., (1995) 273-297
3. Chang, C.C. and Lin, C.J.: LIBSVM: A Library for Support Vector Machines, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
4. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, vol. 2, no. 2, (1998) 121-167
5. Osuna, E., Freund, R. and Girosi, F.: Training Support Vector Machines: an Application to Face Detection, Proc. IEEE Conf. Comp. Vision and Patt. Recog., (1997) 130-136
6. Huang, G.B., Mao, K.Z., Siew, C.K. and Huang, D.S.: Fast Modular Network Implementation for Support Vector Machines, IEEE Trans. Neural Network, vol. 16, (2005)
7. Rowley, H.A., Baluja, S. and Kanade, T.: Neural Network-Based Face Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no.1, (1998) 23-38
8. Kim, H.C., Pang, S., Je, H.M., Kim, D. and Bang, S.Y.: Constructing Support Vector Machine Ensemble, Pattern Recognition, vol. 36, (2003) 2757-2767
9. Stern, H. and Efros, B., Adaptive Color Space Switching for Tracking under Varying Illumination, Image and Vision Computing, vol. 23, (2005) 353-364
10. Wechsler, H., Phillips, P., Bruce, V., Soulie, F. and Huang, T.: Face Recognition: From Theory to Applications, eds. Springer-Verlag (1998)
11. Laws, K.: Textured Image Segmentation, Ph.D. Dissertation, University of Southern California (1980)
12. Viola, P. and Jones, M.J.: Rapid Object Detection using a Boosted Cascade of Simple Features, IEEE Conf. Computer Vision and Pattern Recognition, vol. 1 (2001) 511-518
13. Feraud, R., Bernier, O.J., Viallet, J.E. and Collobert, M.: A Fast and Accurate Face Detection Based on Neural Network, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 1 (2001) 42-53
14. Hsu, R.L., Mohamed, A.M., Jain, A.K.: Face Detection in Color Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no.5 (2002)
15. Gong, S. and McKenna, S. J., psarrou, A.: DYNAMIC VISION: From Images to Face Recognition, Imperial College Press (2000)

16. Singh, S.K., Chauhan, D.S., Vatsa, M. and Singh, R.: A Robust Skin Color Based Face Detection Algorithm, Tamkang Journal of Science and Engineering, vol. 6, no. 4, (2003) 227-234
17. Li, S.Z., Yan, J., Zhang, H.: Learning Illumination-Invariant View Subspaces of Object Appearances, Technical Report, Microsoft Research, Report number MSR-TR-2001-06 (2001)
18. Vapnik, V.N.: The Nature of Statistical Learning Theory, Springer, New York (2000)
19. Tresp, V.: A Bayesian Committee Machine, Neural Comput., vol. 12 (2000)

A New Subspace Analysis Approach Based on Laplacianfaces

Yan Wu and Ren-Min Gu

Dept. of Computer Science and Technology, Tongji University, 1239 Si Ping Road,
Shanghai 200092, P.R. of China
yanwu@mail.tongji.edu.cn

Abstract. A new subspace analysis approach named ANLBM is proposed based on Laplacianfaces. It uses the discriminant information of training samples by supervised mechanism, enhances within-class local information by an objective function. The objective function is used to construct adjacency graph's weight matrix. In order to avoid the drawback of Laplacianfaces' PCA step, ANLBM uses kernel mapping. ANLBM changes the problem from minimum eigenvalue solution to maximum eigenvalue solution, reduces the redundancy of the computing and increases the precision of the result. The experiments are performed on ORL and Yale databases. Experimental results show that ANLBM has a better performance.

1 Introduction

The facial feature extraction is an important step of face recognition. The capability of facial feature extraction directly influences the performance of face recognition. Sub-space analysis is a good method for facial feature extraction. The widely used methods such as Eigenfaces [1] and Fisherfaces [2] are linear dimension reduction methods. However, affected by many complex factors such as expression illumination and pose, the face images should reside on a nonlinear face manifold [3,4,5].

Recently manifold study obtains the people's attention. Manifold is an extension of linear subspace. Manifold study such as LLE [3], ISOMAP [4] and Laplacian Eigen-map [5], are nonlinear dimension reduction methods. However, these methods only are defined on the training set. They can not be used for online dimension reduction of the testing set. So they can not be used for face recognition directly.

He et al [6] proposed a method named LPP (Locality Preserving Projections), which was a linear approximation to Laplacian Eigenmap. He et al [7] used the LPP to do face recognition which was named Laplacianfaces. But it lost a lot of information in PCA step. He et al [8] proposed a method named NPE (Neighborhood Preserving Embedding), which was a linear approximation to LLE. This method also lost a lot of information in PCA step. Cheng et al [9] modified the LPP by kernel mapping, which was named SNLE (Supervised Nonlinear Local Embedding). This method lacked a constructing method for adjacency graph's weight matrix. It can not fully enhance the within-class local information.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 253–259, 2006.

Motivated by [6,7,8,9], a new method named ANLBM is proposed based on Laplacianfaces. It uses the discriminant information of training samples by supervised mechanism, enhances within-class local information by an objective function. The objective function is used to construct adjacency graph's weight matrix. In order to avoid the drawback of the Laplacianfaces' PCA step, ANLBM uses kernel mapping. Many experiments are performed on ORL and Yale database. The experimental results show that, compared with Eigenfaces, Fisherfaces, Laplacianfaces, SNLE, and NPE, ANLBM has a higher recognition rate. In different parameter conditions, ANLBM shows more stable performance.

2 Laplacianfaces

Laplacianfaces is a face subspace analysis method based on LPP. It treats the face image as a point in high-dimensional space. The relationships between face images are treated as the weights between points. Then an adjacency graph can be constructed by these weights. Based on this adjacency graph, a mapping from high-dimensional space to low-dimensional space is defined. This mapping should obtain local structure information of the adjacency graph. Find out this mapping and we can use it for face recognition.

Firstly, a face image, which has r rows and c columns, is represented as a $r \times c$ dimensional vector x_i. The target is to find out a transformation matrix W. It can linearly map vector x_i to a vector y_j in a d-dimensional subspace, here $y_i = W^T x_i$.

2.1 Construct the Adjacency Graph and Find Out the Weight Matrix *S*

$$y = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{t}} & ,x_i \sim x_j \\ 0 & ,\text{otherwise} \end{cases} \tag{1}$$

In (1), $x_i \sim x_j$ means x_i and x_j are adjacent. There are two definitions on adjacency. The first one is that if $\|x_i - x_j\|^2 < \varepsilon$, than $x_i \sim x_j$. But this definition can not ascertain the ε easily. The second one [4] is that x_j belongs to the K nearest points of x_i, also named K-Neighborhood. However, K-Neighborhood is not symmetric. At this time, we can use the following modification: $S_{i,j} = S_{j,i} = min(S_{i,j}, S_{j,i})$ or $S_{i,j} = S_{j,i} = max(S_{i,j}, S_{j,i})$. But it distorts the original local structural information. This is one of the Laplacianfaces' drawbacks.

2.2 Find Out the Transformation Matrix *W* of the Adjacency Graph

In order to obtain the local structural information of the adjacency graph, Laplacianfaces defines an objective function performed by (2). It should be minimized.

$$\frac{1}{2}\sum_{ij}(y_i - y_j)^2 S_{ij} \tag{2}$$

Put $y_i = W^T x_i$ to (2) and transform (2) into (3).

$$W^T X(D - S)X^T W \tag{3}$$

In (3), $X = [x_1, x_2, \ldots, x_n]$. D is a diagonal matrix, $D_{ii} = \sum_j S_{ji}$, $L = D - S$. By Spectral Graph Theory, D_{ii} is the degree of the vertex x_i and L is the Laplacian matrix of the adjacency graph. In order to obtain the local structural information of the adjacency graph and normalize the solution, Laplacianfaces imposes a constraint: $Y^T DY = E$. At last, the problem becomes minimum eigenvalue solution to the generalized eigenvalue problem:

$$X(D - S)X^T w = rXDX^T w \tag{4}$$

Find out the minimum eigenvectors solution $w_1, w_2, \ldots, w_d$, and then $W = [w_1, w_2, \ldots, w_d]$.

2.3 Using W to Do Dimension Reduction

No matter x_i is the training sample or the testing sample, we just use W to do dimension reduction: $y_i = W^T x_i$. However, the number of training samples is always lower than $r \times c$, then XDX^T is a singular matrix. At that time, Laplacianfaces should project the X to a PCA subspace first, and then find out W to do dimension reduction. When the number of training samples is small, the PCA step loses a lot of information. This is just like do Laplacianfaces analysis in Eigenfaces subspace. So when the number of training samples is smaller than $r \times c$, Laplacianfaces can not perform well. This is the second drawback of Laplacianfaces.

3 ANLBM

ANLBM uses a supervised objective function to construct adjacency graph's weight matrix. ANLBM also uses kernel mapping. Thus ANLBM resolves the two drawbacks of Laplacianfaces.

3.1 Construct the Adjacency Graph and Find Out the Weight Matrix S

The weight matrix constructing method of ANLBM is coming from LLE [3]. ANLBM modifies it by adding the supervised mechanism. Formula (5) defines an objective function. It should be minimized.

$$\varepsilon(S) = \sum_i \|x_i - \sum_j S_{i,j} x_j\|^2 \tag{5}$$

Formula (6) defines a constraint of the objective function.

$$\sum_j S_{i,j} = 1 \tag{6}$$

The supervised mechanism is that if x_i and x_j belongs to different class, then set $S_{i,j} = 0$. The details about how to solve the above minimum problem can be found in [3]. In ANLBM, the minimized objective function means every point can be reconstructed by other points in the same class and should minimize the reconstruction error. If x_i and x_j belongs to different class, then set $S_{i,j} = 0$. If x_i and x_j belongs to the same class, $S_{i,j}$ reflects the similarity between x_i and x_j. Treating $S_{i,j}$ as the weight of edge which connects x_i and x_j, then we can get the adjacency graph of ANLBM.

3.2 Find Out the Transformation Matrix *W* of the Adjacency Graph

Firstly, a nonlinear function ϕ is used to map x_i into a higher dimensional vector $\phi(x_i)$. Thus an image set $X = [x_1, x_2, \ldots, x_n]$ is mapped to $\phi(X) = [\phi(x_1), \phi(x_2), \ldots, \phi(x_n)]$. Then dimension reduction can be processed in the higher dimensional space. It is performed as $y_i = \widetilde{W}^T \phi(x_i)$. The higher dimensional space can be treated as the span of $\phi(x_1), \phi(x_2), \ldots, \phi(x_n)$. So there exists a coefficient vector $w = [a_1, a_2, \ldots, a_n]^T$, and $\widetilde{w} = \sum_{i=1}^{n} a_i \phi(x_i) = \phi(X)w$. Thus

$$\begin{aligned}
&\frac{1}{2}\sum_{ij}(y_i - y_j)^2 S_{i,j} \\
&= \frac{1}{2}\sum_{ij}(\widetilde{W}^T \phi(x_i) - \widetilde{W}^T \phi(x_j))^2 S_{i,j} \\
&= \widetilde{W}^T \phi(X)(D - S)\phi(X)^T \widetilde{W}^T \\
&= w^T K(D - S)Kw
\end{aligned} \tag{7}$$

Here $K_{i,j} =< \phi(x_i), \phi(x_j) >$. It is unnecessary to know the nonlinear mapping ϕ explicitly. But in higher dimensional space, the dot product, which is also named kernel function, should be defined clearly. ANLBM uses the Gaussian kernel. So $K_{i,j} = k(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{t}}$. It can also use other kernels. A constraint $Y^T DY = E$, namely, $w^T KDKw = E$, is added to (10). Under the constraint to minimize the objective function equals to find minimum eigenvalue solution to the generalized eigenvalue problem:

$$K(D - S)Kw = rKDKw \tag{8}$$

According to (6), the degree of every vertex is 1, so D is identity matrix. Thus

$$\begin{aligned}
&K(D - S)Kw = rKDKw \\
&KDKw - KSKw = rKDKw \\
&KSKw = (1 - r)KDKw \\
&KSKw = (1 - r)KKw
\end{aligned} \tag{9}$$

K is symmetrical; KSK also is symmetrical; KK is symmetric and positive semidefinite. So the target becomes finding the maximum eigenvalue solution to the generalized eigenvalue problem.

$$KSKw = bKKw \tag{10}$$

Then we can get $W = [w_1, w_2, \ldots, w_d]$. ANLBM changes the problem from minimum eigenvalue solution to maximum eigenvalue solution. It reduces the redundancy of computing and increases the precision of result.

3.3 Using *W* to Do Dimension Reduction

For training set X, the dimension reduction result is $Y = W^T K$. While for testing sample $\widetilde{x}$, the dimension reduction result is represented by (11).

$$\begin{aligned}\widetilde{y} &= W^T \phi(X)^T \phi(\widetilde{x}) \\ &= W^T [\phi(x_1)^T \phi(\widetilde{x}), \phi(x_2)^T \phi(\widetilde{x}), \ldots, \phi(x_n)^T \phi(\widetilde{x})]^T \\ &= W^T [k(x_1, \widetilde{x}), k(x_2, \widetilde{x}), \ldots, k(x_n, \widetilde{x})]\end{aligned} \tag{11}$$

Compared with SNLE, ANLBM uses an objective function to construct adjacency graph's weight matrix, set the degree of every vertex at 1, and enhance the edge information between vertexes. Compared with Laplacianfaces and NPE, supervised mechanism fully uses the discriminant information of training samples; kernel method avoids the drawback of PCA step in Laplacianfaces and NPE.

4 Experimental Results

In order to evaluate the performance of ANLBM, ORL and Yale databases are used. These two databases contain complex expression illumination and pose. The ORL database contains 40 persons and each person has 10 face images. The Yale database contains 15 persons and each person has 11 face images. The two eyes are aligned at the same position and the facial areas are cropped into 32 × 32 pixels. No further preprocessing is done. Figure 1 shows the image samples used for experiment. Suppose the face database has c persons and each person has n images, for each person, randomly choose the k images for training, the remaining $n - k$ images for testing. So it can be $\left(C_n^k\right)^c$ ways to choose the samples. The experiment uses 50 of the $\left(C_n^k\right)^c$ ways to do performance test. The finally result is the mean of the 50 results. Parameter of the kernel function is 1. In this paper, all methods apply nearest-neighbor classifier for its simplicity.

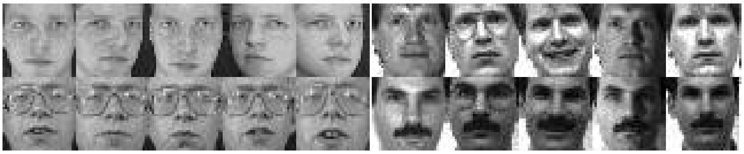


Fig. 1. Preprocessed Face Database

Table 1. The Result (%) on ORL Database

Method	k			
	2	3	4	5
Eigenfaces	70.4(79)	78.8 (119)	84.5 (158)	88.1 (189)
Fisherfaces	72.5(39)	86.2 (39)	91.2 (39)	93.9 (39)
Laplacian-faces	77.8 (40)	86.0 (40)	90.2 (39)	92.7 (40)
NPE	76.9 (40)	83.1 (40)	87.8 (40)	91.6 (46)
SNLE	78.5 (40)	87.5 (40)	91.9 (40)	94.5 (40)
ANLBM	78.9 (53)	87.6 (41)	92.1 (40)	94.7 (42)

Table 2. The Result (%) on Yale Database

Method	k			
	2	3	4	5
Eigenfaces	45.9 (29)	51.8 (44)	54.8 (59)	58.1 (74)
Fisherfaces	41.1 (14)	60.7 (14)	68.5 (14)	74.1 (14)
Laplacian-faces	53.0 (16)	64.2 (15)	69.7 (15)	75.1 (15)
NPE	53.6 (15)	63.7 (16)	70.5 (20)	75.3 (22)
SNLE	51.5 (15)	63.6 (15)	70.6 (20)	74.8 (19)
ANLBM	54.3 (15)	66.1 (16)	72.9 (17)	77.0 (18)

Table 1 and table 2 show the recognition rates on ORL and Yale database respectively. The values in the parentheses are the corresponding dimensionality of the best result. The results in table 1 and table 2 show that ANLBM can get higher recognition rate in different face database and different training numbers. However, using Eigenfaces as baseline, Fisherfaces Laplacianfaces NPE and SNLE don't have absolute advantage in different face databases and different training numbers. When the number of training samples is small, ANLBM has more advantage. Compared with Eigenfaces Fisherfaces Laplacianfaces SNLE and NPE, ANLBM has higher recognition rate. In different parameter condition, ANLBM shows more stable performance.

5 Conclusions

Based on Laplacianfaces, this paper proposes a new face subspace analysis method named ANLBM. It uses the discriminant information of training samples. ANLBM uses an objective function to enhance local structural information. In order to avoid the drawback of the Laplacianfaces' PCA step, ANLBM uses kernel mapping. ANLBM changes the problem from minimum eigenvalue solution to maximum eigenvalue solution. It reduces the redundancy of computing and increases the precision of result. The experiments, which are performed on ORL and Yale databases, show that ANLBM has an impressive performance.

References

1. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience, Vol. 3, No. 1 (1991) 71-86
2. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Trans. Pattern Analysis Mach. Intel., Vol. 19, No. 7 (1997) 711-720
3. Roweis, S.T., Saul, L.K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science, Vol. 290, No.22 (2000) 2323-2326
4. Tenenbaum, J.B., Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science, Vol. 290, No. 12 (2000) 2319-2323
5. Belkin, M., Niyogi, P.: Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. Proc. Conf. Advances in Neural Information Processing System (2002)
6. He, X., Niyogi, P.: Locality Preserving Projections, Proc. Conf. Advances in Neural Information Processing Systems (2003)
7. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.: Face Recognition Using Laplacianfaces. IEEE Trans. Pattern Analysis Mach. Intel., Vol. 27, No. 3 (2005) 328-340
8. He, X., Cai, D., Yan, S., Zhang, H.: Neighborhood preserving embedding. Computer Vision, ICCV, Vol. 2, No. 10 (2005) 1208-1213
9. Cheng, J., Liu, Q., Lu, H., Chen, Y.: A supervised nonlinear local embedding for face recognition. Image Processing, ICIP, Vol. 10, No.1 (2004) 83-86

Rotation Invariant Face Detection Using Convolutional Neural Networks

Fok Hing Chi Tivive and Abdesselam Bouzerdoum*

School of Electrical, Computer and Telecommunications Engineering,
University of Wollongong, Northfields Avenue, Wollongong, NSW 2522,
Australia
tivive@uow.edu.au, a.bouzerdoum@ieee.org

Abstract. This article addresses the problem of rotation invariant face detection using convolutional neural networks. Recently, we developed a new class of convolutional neural networks for visual pattern recognition. These networks have a simple network architecture and use shunting inhibitory neurons as the basic computing elements for feature extraction. Three networks with different connection schemes have been developed for in-plane rotation invariant face detection: fully-connected, toeplitz-connected, and binary-connected networks. The three networks are trained using a variant of Levenberg-Marquardt algorithm and tested on a set of 40,000 rotated face patterns. As a face/non-face classifier, these networks achieve 97.3% classification accuracy for a rotation angle in the range $\pm 90^0$ and 95.9% for full in-plane rotation. The proposed networks have fewer free parameters and better generalization ability than the feedforward neural networks, and outperform the conventional convolutional neural networks.

1 Introduction

The problem of invariant recognition has been a challenging task for computer vision community, as in practice, perfect invariance is very difficult to achieve, because of the computation inaccuracies and the continuous nature of some transformations [20]. Many algorithms have been proposed, which can be grouped into three categories, namely integral invariance, algebraic invariance and neural networks [20]. In integral and algebraic approaches, the input space is transformed into another space such that the features extracted from the latter are invariant to some geometric transformations. Neural approaches, on the other hand, attempt to build invariance through learning, by often combining the feature extraction stage with the classification stage to achieve invariant recognition.

A simple integral approach is the Fourier transform, which is used to transform a pattern from the spatial domain into the frequency domain; the magnitude of the frequency spectrum is invariant to translation. More advance integral and algebraic methods such as Fourier-Mellin integral [6] and moment functions [11,17] have been developed to define a set of descriptors that are invariant to rotation,

* *Senior Member, IEEE.*

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 260–269, 2006.

translation, and scaling transformations. However, these invariant functions have their own drawbacks. For instance, the Fourier-Mellin integral does not converge in the general case, but only under certain strong conditions, and the computation of the Fourier-Mellin descriptors is costly [16]. The geometric moments suffer from a high degree of information redundancy [7], and are sensitive to noise — these problems have been investigated by many researchers [18, 12, 10].

On the other hand, artificial neural networks have some desirable characteristics: (i) they are fault tolerance learning machines; and (ii) they can acquire knowledge from the input data through learning. Barnard and Casasent [3] mentioned three strategies to incorporate invariance into a neural network model, i.e., invariant feature space, invariance by training and invariance by structure. The first strategy consists of two stages. First, the input pattern is mapped into an invariant feature space, and second, the extracted features are used as inputs to a neural network classifier. This approach, however, requires prior knowledge of the problem in order to design the invariant feature detector. The second strategy is to train a neural network with different exemplars of the same object. These exemplars are generated by applying the transformation on the object itself, i.e., different aspect views. Provided that there are sufficient training exemplars and the network learns them properly, the trained network can be expected to generalize correctly to the different orientations of the object. The last strategy is by structuring the network architecture appropriately, e.g., wiring the first few layers of the network in such a way that the network can learn to extract invariant features. This type of networks is known as *convolutional neural networks* (CoNNs) which is derived from the understandings of mammalian's visual cortex. Fukushima *et al.* [8] were the first developed a CoNN called the *Neocognitron*, which has shown to possess a certain degree of invariance. LeCun *et al.* [14], on the other hand, proposed a series of convolutional neural network architectures, dubbed LeNet (1-5), for optical character recognition, in which the input characters are often subject to geometric distortions. The problem of these two-dimensional (2-D) networks is that the network architecture is complex with a large number of trainable parameters.

Recently, we have developed a new class of convolutional neural networks known as *SICoNNets* that has a simple network architecture and consists of a special computing element, the *shunting inhibitory neuron*, for feature extraction [19]. The motivation of using this type of neurons rather than sigmoid type is that the neuron is based on the shunting inhibitory mechanism, which has been used to model a number of visual and cognitive functions [9]. Contrary to a sigmoid neuron, a single shunting inhibitory neuron can solve linearly nonseparable classification problems by forming nonlinear decision boundaries [4,5]. Moreover, when the shunting inhibitory neuron is applied in other neural network models for supervised pattern classification and regression, it has been shown to be more powerful than the sigmoid neuron or perceptron [2].

In this paper, we employ SICoNNets for rotation invariant face detection. Here, rotation invariance is achieved by training the networks on rotated face patterns and non-face patterns. To determine how well the SICoNNets perform compared to other neural models, the multilayer perceptrons (MLPs) and the early model of CoNNs, LeNet [13], have been trained and tested on the same classification task. The next section describes the architecture of the SIConNet and its basic computing element, the shunting inhibitory neuron. Section 3 presents the rotation invariant face classifier and describes the network training and testing procedure. Experimental results and performance analysis are given in Section 4. Finally, Section 5 presents concluding remarks.

2 SICoNNet Architecture

The SICoNNet architecture is a multilayer network based on the three structural concepts of LeNet-5 [14]: local receptive fields, weight sharing and sub-sampling. The input layer is a 2-D plane of arbitrary size, acting as the network retina to receive inputs from the environment. After the input layer, there are several hidden layers in which the neurons are arranged into several planes called *feature maps*. Each neuron in a feature map is connected locally to a small neighborhood (receptive field) in the input image, and each hidden layer has its own receptive field size. In other words, a receptive field is a small $N \times N$ (N is an odd integer) region of the input image whence the neuron receives its inputs. In a feature map, all the neurons share the same set of weights (weight sharing) to connect to their receptive fields to cover the entire input plane. The mechanisms of local connection and weight sharing constrain each neuron in a feature map to perform the same computation operation on different parts of the input image; that is, the same elementary visual feature is extracted from different positions in the input plane. Other feature maps in that layer perform the same operation with different sets of weights to extract different types of local features. Instead of having a sub-sampling layer after the convolutional layer as in LeNet-5 (i.e., a convolutional layer followed by a sub-sampling layer), the sub-sampling operation is incorporated into each hidden layer. This is done by shifting the centers of receptive fields of adjacent neurons by two positions in the vertical and horizontal directions. Hence, the network structure has fewer hidden layers and connections, and the size of the feature maps are reduced by one quarter in successive layers. To reduce the number of trainable weights between the last hidden layer and the output layer, a local averaging operation is performed on all the feature maps; that is, small 2×2 non-overlapping regions are averaged, and the resulting signals are fed into the output layer. However, if the feature maps of the last hidden layer consist of single neurons, the outputs of these neurons serve as inputs directly to the output layer.

To avoid hand-coding the connections between layers, three systematic connection schemes are developed for our CoNNs: full-connection, toeplitz-connection and binary-connection. In a full-connection scheme, each feature map is fully connected to all feature maps of the succeeding layer, and each hidden layer can have an arbitrary number of feature maps. For the binary-connection scheme, Fig. 1(a), each feature map branches out to two feature maps in the succeeding layer similar to a binary tree, whereas in the toeplitz-connection scheme,

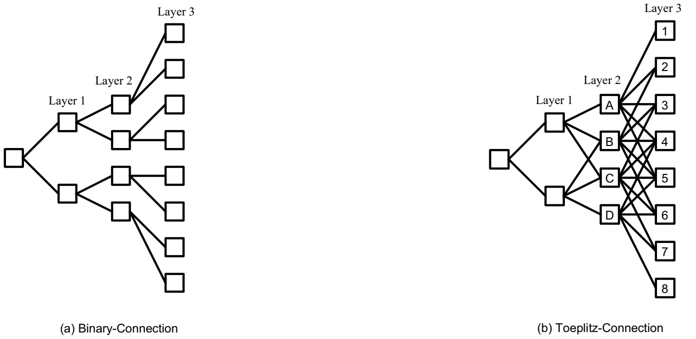


Fig. 1. The partial-connection schemes: (a) binary-connection and (b) toeplitz-connection

L3 Feature Map	Connections from L2 to L3				
1	A				
2	B	A			
3	C	B	A		
4	D	C	B	A	
5		D	C	B	A
6			D	C	B
7				D	C
8					D

Fig. 2. Connections between feature maps in L2 and L3 of the toeplitz architecture

Fig. 1(b), each feature map may have one-to-one or one-to-many links with feature maps of the preceding layer, forming a toeplitz connection matrix. As an example of the toeplitz-connection scheme, Table 2 illustrates the connections between Layer 2 (L2) and Layer 3 (L3). Suppose L3 contains eight feature maps, labelled 1 to 8 (first column), and L2 has four feature maps, labelled A to D. Feature maps 1 and 8 have one-to-one connections with feature maps A and D, respectively. Feature map 2 has connections with feature maps A and B, whereas feature map 3 has connections with feature maps A, B and C. The rest of the connections form a *Toeplitz* matrix, hence the name.

The feature maps of the SICoNNet are made up of shunting inhibitory neurons, and the neural activity at location (i, j) in the feature map $\{L, k\}$ is expressed as

$$Z_{L,k}(i,j) = \frac{f_L\Big(\sum_{m=1}^{S_{L-1}} [C_{L,k} * Z_{L-1,m}]_{(2i)(2j)} + b_{L,k}(i,j)\Big)}{a_{L,k}(i,j) + g_L\Big(\sum_{m=1}^{S_{L-1}} [D_{L,k} * Z_{L-1,m}]_{(2i)(2j)} + d_{L,k}(i,j)\Big)}, \qquad (1)$$

$$\forall \;\; i,j = 1, ..., M_L$$

where $*$ denotes 2-D convolution, the parameters $C_{L,k}$ and $D_{L,k}$ are the set of trainable weights, $b_{L,k}$ and $d_{L,k}$ are the biases, $a_{L,k}$ is the passive decay term, f_L and g_L are the activation functions, and M_L is the size of the feature map at the Lth layer. All the shunting neurons in a feature map share the same set of weights $C_{L,k}$ and $D_{L,k}$, the bias parameters, and the passive decay rate term. In order to avoid dividing by zero in (1), the decay parameter is constrained as follows:

$$\left[a_{L,k}(i,j) + g_L\Big(\sum_{m=1}^{S_{L-1}} [D_{L,k} * Z_{L-1,m}]_{(2i)(2j)} + d_{L,k}(i,j)\Big)\right] \geq \varepsilon > 0, \qquad (2)$$

and this condition is imposed during both initialization and training processes. The outputs of the network are generated with sigmoid type or linear neurons. The response of a sigmoid neuron is the weighted sum of its input signals, plus a bias term, passed through an activation function; mathematically, it is given by

$$y = h\Big(\sum_{i=1}^{S_N} w_i z_i + b\Big), \qquad (3)$$

where y is the response of the neuron, h is the activation function, w_i's are the connection weights, z_i's are the inputs to the sigmoid neuron, S_N is the number of input signals, and b is the bias term. Figure 3 shows a schematic diagram of a toeplitz-connected SICoNNet with four feature maps in the hidden layers. After the local averaging operation, all the outputs signals from the feature maps of layer 2 are fully connected to the perceptron.

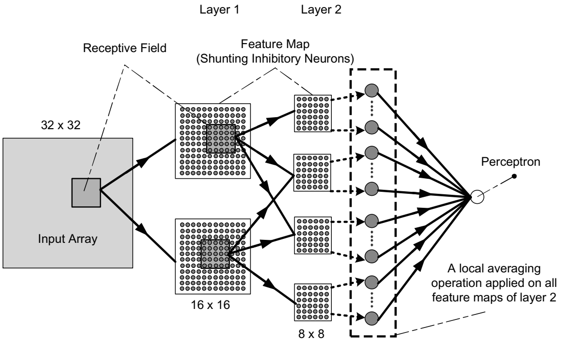


Fig. 3. A schematic diagram of a toeplitz-connected SICoNNet

3 Rotation Invariant Face Classifier

To develop a face/non-face classifier that can be used for rotation invariant face detection, the proposed CoNN is designed into a three layer network (two hidden layers and one output layer) that accepts input image of size 32×32 and

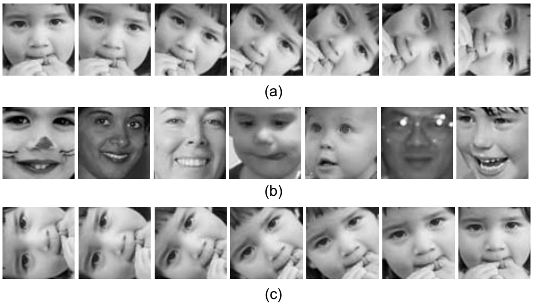


Fig. 4. Face patterns: (a) face rotated in the range $[0^0, -90^0]$, (b) quasi-frontal faces, and (c) face rotated in the range $[90^0, 0^0]$

produces a single output signal. The first hidden layer, L1, has two feature maps and the second hidden layer, L2, has four feature maps. After some preliminary experimentations, the activation functions f_L and g_L in L1 were chosen as the hyperbolic tangent and exponential functions, respectively, whereas in L2, g_L is the logarithmic sigmoid function. For the output layer, h is the linear activation function. To allow more overlapping input information processed by the shunting inhibitory neurons which may increase the degree of invariance in the network, different size of receptive field, ranging from 5×5 to 9×9, is used. Moreover, three networks are developed according to the systematic connection schemes and trained with a variant of Levenberg-Marquardt algorithm proposed by Ampazis and Perantonis [1]. As for MLPs, two network structures are implemented, i.e., a two layer and a three layer MLPs with different number of neurons in each hidden layer, varying from 5 to 50 neurons. All the MLPs have one perceptron at the output layer, and they are trained with the scale conjugate gradient training algorithm.

A large database of rotated and quasi-frontal face patterns has been generated for training and testing the CoNNs as a rotation invariant face classifier. The quasi-frontal face patterns were taken from the face database created by Phung *et al.* [15], which contains face images with people of different ages, ethnic backgrounds, and different lighting conditions. The rotated face patterns were generated by in-plane rotating and cropping face images at different angles, in the range $\pm 90^0$ with steps of 15^0. These face images were collected from different sources on the Web varying in terms of the background, the people, and the illumination condition. To obtain face patterns beyond that range, every face pattern is folded along the X-axis direction. Some examples of rotated face patterns are shown in Fig. 4. The proposed CoNN is trained and tested for partial and full rotation invariance; that is, after training, the network can classify rotated face patterns in the range $[90^0, -90^0]$ and $[0, 360^0]$, respectively. To train a face classifier that can discriminate rotated face patterns, the training set consists of 2000 quasi-frontal face patterns, 4000 rotated face patterns and 6000 non-face patterns. Another training set with 12,000 face patterns is generated to cover a 360^0 rotation range. It includes frontal face patterns, rotated face patterns and their folded counterparts. For evaluation, two test sets are prepared. The first test set has 20,000 rotated faces, and the second test set has 40,000 faces patterns where every rotated face pattern is folded along the X-axis. In every test set, the nonface patterns are obtained from a bootstrap procedure. The patterns in the training and test sets are normalized by scaling linearly every image pixel to the range $[-1, 1]$. Finally, the desired outputs corresponding to face and non-face patterns are labelled as 1 and -1, respectively. The same training and test sets are used for the MLP and LeNet networks.

4 Experimental Results and Performance Analysis

The first test is to determine the network generalization ability of discriminating rotated faces in the range $\pm 90^0$ from segmented non-face patterns. If the trained network yields a high classification rate on the test sets, the network is considered not only to have the ability to discriminate between face and non-face patterns, but also invariant to in-plane rotation. Therefore, each network has been trained and tested three times, and the best classification result has been recorded among the three trials. Tables 1 presents the classification performances of the CoNNs. The high classification accuracies obtained in Table 1 show that the proposed networks can be trained to be partially rotation invariant. Based on the four different combinations of receptive fields sizes, all the networks achieve classification accuracies over 93% with the best performance obtained from a toeplitz-connected network with a classification rate of 97.2% for face and 97.3% for non-face patterns. Across all four sets of receptive fields sizes, the toeplitz-connected network has the highest average performances with accuracies of 95.8%. On average the partially-connected CoNNs outperform the fully-connected network. One possible reason is that the latter has too many connections, and each feature map is compelled to extract an average feature from all feature maps of the preceding layer.

All the MLPs have been evaluated on the same test set, and Table 2 presents the classification performances of the two best trained MLPs for each network structure. The experimental results show that MLPs perform poorly when the input data are subject to variations in rotation. With a two layer MLP, the highest classification accuracy is 88%, and for a three layer network, the highest performance is 88.9%. In terms of number of trainable weights, the two layer

Table 1. Classification rates of the binary-, toeplitz-, and full-connected SICoNNets tested on rotated face patterns in the range $\pm 90^0$ based on different sizes of receptive fields

Receptive Field		Binary-Net (%)			Toeplitz-Net (%)			Full-Net (%)		
L-1	L-2	Face	NFace	Acc.	Face	NFace	Acc.	Face	NFace	Acc.
5 × 5	5 × 5	95.5	94.7	95.1	97.2	97.3	97.3	96.3	96.0	96.2
7 × 7	5 × 5	96.6	96.0	96.3	96.2	93.6	94.9	95.2	92.4	93.8
9 × 9	5 × 5	94.6	94.6	94.6	95.1	95.1	95.1	94.8	94.1	94.5
9 × 9	7 × 7	96.2	95.7	96.0	96.0	95.9	96.0	95.8	96.8	96.3

Table 2. Classification performances of the two best trained MLPs tested on the rotated face patterns in the range ±90^0

Net Index	Number of neurons		Face (%)	Non-Face (NFace) (%)	Accuracy (Acc.) (%)
	Layer 1	Layer 2			
Net-01	40	0	86.2	89.9	88.0
Net-02	45	0	87.5	87.8	87.7
Net-03	40	30	89.4	88.5	88.9
Net-04	35	50	87.5	89.8	88.7

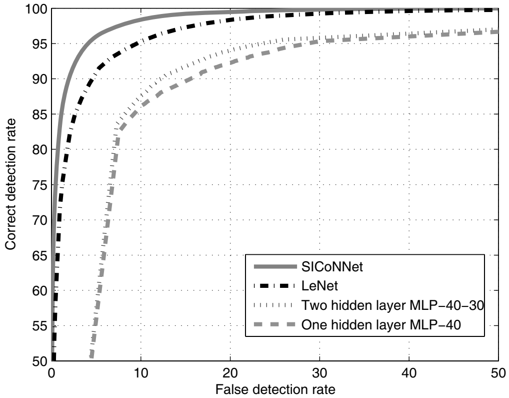


Fig. 5. ROC curves of three different networks - SICoNNet, LeNet and MLPs, tested on rotated face patterns in the range ±90^0

MLP (Net-01) has 41,041 trainable parameters, whereas the three layer MLP (Net-03) has 42,261 weights. Most of the trainable parameters are the weights between the input and the first hidden layer since every neuron in the first hidden layer has 1024 weights. In comparison to MLPs, the CoNNs have better classification performance and fewer number of trainable network parameters; for instance, using receptive fields size of 5×5 in the hidden layers, the three layer toeplitz-connected network has 383 weights with a correct face classification rate of 97.2%.

Figure 5 shows the *Receiver Operating Characteristic* (ROC) curves of the MLP, LeNet and SICoNNet when trained and tested on rotated face patterns in the range ±90^0. In this experiment, the LeNet and SICoNNet have similar network structure with receptive fields sizes of 9×9 and 7×7, using a full-connection scheme and a local averaging operation at the last hidden layer. However, the main difference is that the feature maps of the LeNet consist of sigmoid neurons, whereas those of the SICoNNet contain shunting inhibitory neurons. Among the three networks, the SICoNNet has the best correct detection rate, followed by LeNet and then MLPs. The experimental result indicates that not only does the network structure affect the classification performance, but

Table 3. Classification rates of the binary-, toeplitz-, and full-connected SICoNNets tested on face patterns rotated in the range of 360^0 based on different sizes of receptive fields

Receptive Field		Binary-Net (%)			Toeplitz-Net (%)			Full-Net (%)		
L-1	L-2	Face	NFace	Acc.	Face	NFace	Acc.	Face	NFace	Acc.
5×5	5×5	94.6	92.8	93.7	94.4	94.4	94.4	94.4	93.0	93.7
7×7	5×5	93.1	90.3	91.7	93.4	90.5	92.0	91.9	90.9	91.4
9×9	5×5	93.2	94.0	93.6	94.8	93.4	94.1	92.0	92.0	92.0
9×9	7×7	96.1	95.6	95.9	95.8	95.4	95.6	96.1	94.5	95.3

also the type of neuron (sigmoid or shunting inhibitory neuron) used to extract the features from the input patterns.

The last experiment is to evaluate the SICoNNets for full rotation invariance. Therefore, all three network architectures are trained on the second training set and evaluated on the test set containing 40,000 rotated face patterns; their classification rates are listed in Table 3. Among the four combinations of receptive fields, the last combination (i.e., 9×9 and 7×7) yields the highest classification accuracy with 95.9% using a binary-connected network, followed by the toeplitz-connected network (95.6%) and fully-connected network (95.3%).

5 Conclusion

This paper investigates the problem of rotation invariant face detection using a class of shunting inhibitory convolutional neural networks (SICoNNets). Training algorithms have been developed for three different architectures: binary-connected network, toeplitz-connected network and fully-connected network. All three CoNNs can learn rotation invariance very efficiently. As a face/non-face classifier, the proposed network achieves a classification accuracy of 97.3% for in-plane rotation in the range $\pm 90^0$, and using receptive fields of sizes 9×9 and 7×7, a classification accuracy of 95.9% is achieved for full in-plane rotation. This demonstrates that SICoNNets can be applied in rotation invariant face detection system. Experimental results show that traditional convolutional neural networks, which use sigmoid neurons for feature extraction do not perform as well as SICoNNets. For comparison purposes, multilayer perceptrons have also been trained for rotation invariant face detection. It was found that MLPs do not perform as well as CoNNs, in general. Furthermore, the MLP networks possess a large number of weights, which makes them more prone to over-fitting the training data.

References

1. N. Ampazis and S. J. Perantonis. Two highly efficient second-order algorithms for training feedforward networks. *IEEE Transactions on Neural Networks*, 13(5):1064–1074, 2002.

2. G. Arulampalam and A. Bouzerdoum. Application of shunting inhibitory artificial neural networks to medical diagnosis. In *Proc. of the Seventh Australian and New Zealand Intelligent Information Systems Conference*, pages 89–94, Perth, 2001.
3. E. Barnard and D. Casasent. Invariance and neural nets. *IEEE Transactions on Neural Networks*, 2:498–508, 1991.
4. A. Bouzerdoum. A new class of high-order neural networks with nonlinear decision boundaries. In *Proc. of the Sixth International Conference on Neural Information Processing*, volume 3, pages 1004–1009, Perth, 1999.
5. A. Bouzerdoum. Classification and function approximation using feed-forward shunting inhibitory artificial neural networks. In *Proc. of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, pages 613–618, 2000.
6. D. Casasent and D. Psaltis. Position, rotation and scale-invariant optical correlation. *Applied Optics*, 15(7):1795–1799, 1976.
7. C.-W. Chong, P. Raveendran, and R. Mukundan. Translation invariants of zernike moments. *Pattern Recognition*, 36(8):1765–1773, 2003.
8. K. Fukushima, S. Miyake, and T. Ito. Neocognitron: a neural network model for a mechanism of visual pattern recognition. *IEEE Transactions Systems, Man, and Cybernetics*, SMC-13(5):826–834, 1983.
9. S. Grossberg, editor. *Neural Networks and Natural Intelligence.* MIT Press, Cambridge, Massachusetts, 1988.
10. M. Gruber and K. Y. Hsu. Moment-based image normalization with high noise tolerance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):136–139, 1997.
11. M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions Information Theory*, IT-8:179–187, 1962.
12. A. Khotanzad and J. H. Lu. Classification on invariant image representations using a neural network. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38:1028–1038, 1990.
13. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.
14. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998.
15. S. L. Phung, A. Bouzerdoum, and D. Chai. Skin segmentation using color pixel classification: analysis and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):148 – 154, 2005.
16. M. A. Rodrigues. *Invariants for pattern recognition and classification*, volume 42 of *machine perception and artificial intelligence.* World Scientific, Singapore, 2000.
17. M. Teague. Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70(8):920–930, 1980.
18. C. H. Teh and R. T. Chin. On image analysis by the method of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):496–513, 1988.
19. F. H. C. Tivive and A. Bouzerdoum. Efficient training algorithms for a class of shunting inhibitory convolutional neural networks. *IEEE Transactions on Neural Networks*, 16(3):541– 556, 2005.
20. J. Wood. Invariant pattern recognition: a review. *Pattern Recognition*, 29(1):1–17, 1996.

Face Tracking Algorithm Based on Mean Shift and Ellipse Fitting

Jianpo Gao, Zhenyang Wu, and Yujian Wang

Department of Radio Engineering, Southeast University, 210096 Nanjing, China
zywu@seu.edu.cn

Abstract. The mean shift algorithm is an efficient technique for object tracking. However, it has a shortcoming that it can't adjust scale with object during tracking process. There are presently no effective ways to solve this problem. The kernel bandwidth of mean shift tracker in one frame is generally steered by the object scale obtained in the previous frame, so it is very important for mean shift tracker to correctly describe the scale of the target in very frame. In accordance with the kernel-bandwidth effect on the mean shift tracker and the property of face, this paper introduces a new idea that uses direct least square ellipse fitting to adjust the facial scale. The experimental results demonstrate the efficiency of this algorithm. Its performance has been proven superior to the original mean shift tracking algorithm.

1 Introduction

Face detection and tracking, as the first process of facial information management, attaches more and more attention in the computer vision field. Nowadays, face detection and tracking in video sequence can find its place in many application, such as video conferencing, automatic surveillance, digital video management, etc..

Face tracking in video sequence is based on dynamic image processing, its fundamental task is to capture the location and scale of face in video sequence. For video image, there is so much information can be used as cues of face tracking, such as color, movement, shape and texture. Each of these cues has its respective character and applying condition. As far as the color image is considered, different objects always cluster into different color regions, so we can use color information to detect distinct object. Skin color is an important feature for face and used widely by many face tracking system [1,2,4]. Color histogram, as a key tool to describe color feature, has advantage that it is insensitive to rotation, transform and scale change of object. Dorin Comaniciu and others proposed an object tracking algorithm based on mean shift which use the color histogram as cue [4]. This method has been proved simple and efficient. However, it has a shortcoming that it can't adapt scale with object during tracking process. There are presently no effective ways to solve this problem.

In accordance with the fact that mean shift tracker always can give the right target location when a larger kernel bandwidth is chosen and the facial shape can be appropriate to an ellipse, this paper introduces a new idea that uses direct least square ellipse fitting to adjust the facial scale based on the mean shift tracking framework. Fig.1 gives the schematic illumination of our algorithm. The experimental results

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 270 – 277, 2006.

demonstrate the efficiency of this algorithm. Its performance has been proven superior to the original mean shift tracking algorithm.

The rest of this paper is organized as follows. Section 2 analyses the face-tracking algorithm based on mean shift. Section 3 presents our face shape describing method using direct least squares ellipse fitting. Section 4 gives experimental results to demonstrate the efficiency of our algorithm. Conclusion is given in the last section.

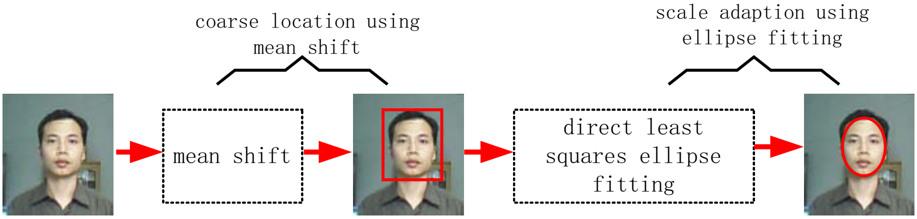
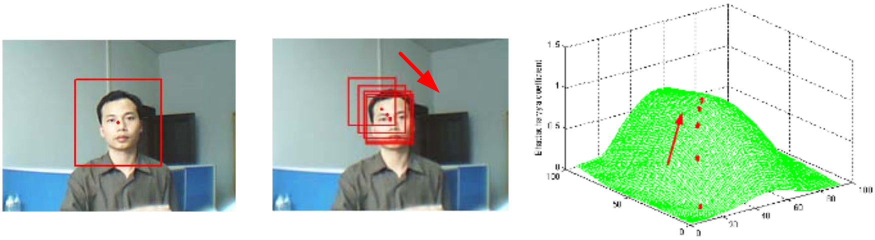

coarse location using mean shift
scale adaption using ellipse fitting
mean shift
direct least squares ellipse fitting

Fig. 1. The schematic illumination of our algorithm

2 Face Tracking Based on Mean Shift

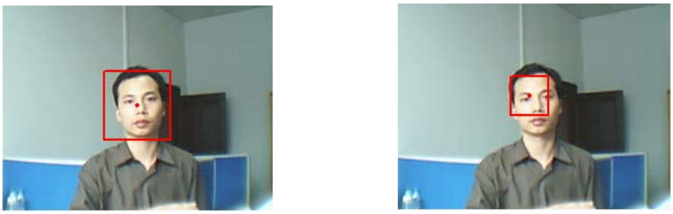
The mean shift algorithm [4,5,6] is an effective method for mode seeking in probability space, which is based on the theory of nonparametric kernel probability density estimation. Dorin Comaniciu , etc proposed a simple and efficient object tracking algorithm [4] based on the mean shift theory. Because the mean shift trakcer uses the color histogram as tracking feature, it is robust to the transform, rotation and scale variability of target. What more, this tracking algorithm is simple and efficient. Just the above advantages make this algorithm attract more and more attention in computer vision field. Fig.2 (b) shows the face seeking process from the initial location in one frame. Fig.2 (c) shows all values of the bhattacharyya coefficient corresponding to the rectangle marked in Fig.2 (a), the locations during the mean shift iterations are also shown. From Fig.2 we can see clearly the realization mechanism of face tracking using mean shift method.

From the mean shift theory, it is clear that the mean shift algorithm fulfills the only task seeking model of probability density, so in fact mean shift tracker can only locate target and it can't give us the accurate scale of target. In general, we always choose an empirical kernel bandwidth according to the prior scale knowledge of target firstly, then we finish a mean shift tracking process using this fixed kernel bandwidth. However, it is difficult to estimate the true scale of target before tracking, so the kernel bandwidth chosen by means of the above method can't avoid different from the true target scale. The difference between the kernel bandwidth and the true target scale can results in the degeneracy of tracking effect. In general, when the background is not too complex, we can always get an accurate location of target if a larger scale (bandwidth) is selected during tracking process, but the target scale in not accurate (larger than true scale) on this condition. This can be seen clearly from Fig.3 (a). On the contrary, from Fig.3 (b) we can see, if we select a smaller scale (bandwidth), then we can get neither the accurate target location because of the local extremums nor the accurate target scale.



(a) (b) (c)

Fig. 2. The realization mechanism of face tracking using mean shift. (a) A face picture. (b) The face seeking process from the initial location in one frame. (c) All values of the bhattacharyya coefficient corresponding to the rectangle marked in Fig.2 (a).



(a) (b)

Fig. 3. The demonstration of bandwidth effect for face tracking based on mean shift. (a) The tracking result with a larger scale (bandwidth). (b) The tracking result with a smaller scale (bandwidth).

During the mean shift tracking process, the kernel bandwidth of the t frame is always chosen according to the target scale attained from the tracking result in $t-1$ frame, so attaining the precise target scale in every frame is important, because the wrong estimation of target in one frame can cause tracking failure in the next frames. However, There are presently no effective ways to adjust the target scale or kernel bandwidth during the mean shift tracking. In the literature [4], the target scale adjusting method is to track the target with the kernel bandwidth that is 1, 1.1 and 0.9 times as the last kernel bandwidth respectively, then choose the one tracking result that product the largest Bhattacharyya coefficient. But this method can't solve the above target scale problem perfectly.

From the above analysis, we know that it can always attain the right location of the face target during the mean shift tracking process if a little larger kernel bandwidth is chosen, but the attained target scale is little larger than the true one. In general, facial shape can be approximate to an ellipse, so if we can describe the elliptical shape of face accurately on the base of approximate location given by mean shift tracker with a little larger kernel bandwidth, and using the attained elliptical shape directs the choice of kernel bandwidth in the next frame. Then the scale problem of mean shift tracker can be solved. Now the key problem is how to describe the elliptical shape of face efficiently and accurately. The above consideration just is the source of the new idea that uses direct least square ellipse fitting to adjust the facial scale based on the mean shift tracker proposed in this paper.

3 Describe the Face Shape Using Ellipse Fitting

In computer vision, there are two common methods used to ellipse fitting. One is hough transform based on vote mechanism, the other is least squares technique. Andrew Fitzgibbon, etc. proposed an ellipse fitting algorithm by means of direct least squares method [3]. This algorithm is robust and efficient. In this paper, we use it to describe the elliptical shape of face during the mean shift tracking process.

A general conic by an implicit second order polynomial can be represented as

$$\mathrm{F}\,(\mathbf{a},\mathbf{x}) = \mathbf{a}^{\mathrm{T}}\mathbf{x} = ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{1}$$

where $\mathbf{a} = [a \quad b \quad c \quad d \quad e \quad f]^{\mathrm{T}}$, $\mathbf{x} = [x^2 \quad xy \quad y^2 \quad x \quad y \quad 1]^{\mathrm{T}}$.

Assuming $\mathrm{F}(\mathbf{a},\mathbf{x_i})$ denotes the algebraic distance of point (x_i, y_i) to the conic $\mathrm{F}\,(\mathbf{a},\mathbf{x}) = 0$. The fitting of general conic can be solved by means of seeking coefficient vector $\mathbf{a}$ that minimizes the equation (2).

$$E(\mathbf{a}) = \sum_{i=1}^{N} \mathrm{F}(\mathbf{a},\mathbf{x_i})^2 = \|\mathbf{Da}\|^2 \tag{2}$$

where $\mathbf{D} = [\mathbf{x}_1 \quad \mathbf{x}_2 ... \mathbf{x}_N]^{\mathrm{T}}$, $\mathbf{a} = [a \quad b \quad c \quad d \quad e \quad f]^{\mathrm{T}}$, $\mathbf{x_i} = [x_i^2 \quad x_i y_i \quad y_i^2 \quad x_i \quad y_i \quad 1]^{\mathrm{T}}$, N is the number of data that used ellipse fitting.

It is clear that equation (1) denotes ellipse when $b^2 - 4ac < 0$, since $\mathbf{a} = [a \quad b \quad c \quad d \quad e \quad f]^{\mathrm{T}}$ is a free parameter, so we can use equality constraint $4ac - b^2 = 1$ instead of $b^2 - 4ac < 0$.

Assuming

$$C = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

Then the equality constraint $4ac - b^2 = 1$ can be denoted as following equation in the form of matrix.

$$\mathbf{a^T C a} = 1 \tag{4}$$

Now the above least squares ellipse-fitting problem can be written as

$$\mathbf{a} = \arg\min_{\mathbf{a}} \|\mathbf{Da}\|^2 \tag{5a}$$

$$\mathbf{a^T C a} = 1 \tag{5b}$$

Introducing the Largrange multiplier λ and differentiating , we can get

$$\begin{cases} 2\mathbf{D}^T\mathbf{Da} - 2\lambda\mathbf{Ca} = 0 \\ \mathbf{a}^{\mathrm{T}}\mathbf{Ca} = 1 \end{cases} \tag{6}$$

equation (6) can be rewritten as

$$\mathbf{Sa} = \lambda\mathbf{Ca} \tag{7a}$$

$$\mathbf{a}^{\mathrm{T}}\mathbf{Ca} = 1 \tag{7b}$$

where $S = \mathbf{D}^{\mathrm{T}}\mathbf{D}$, solving the above equation by means of generalized eigenvectors method, we can get

$$\mathbf{a} = \left(\sqrt{\frac{1}{\mathbf{u}_i^T\mathbf{Cu}_i}}\right)\mathbf{u}_i = \left(\sqrt{\frac{\lambda_i}{\mathbf{u}_i^T\mathbf{Su}_i}}\right)\mathbf{u}_i \tag{8}$$

where λ_i and $\mathbf{u}_i$ are the eigenvalue and eigenvector of equation (7a) , respectively. It can be proved that the solution of $\mathbf{a}$ is unique [3].

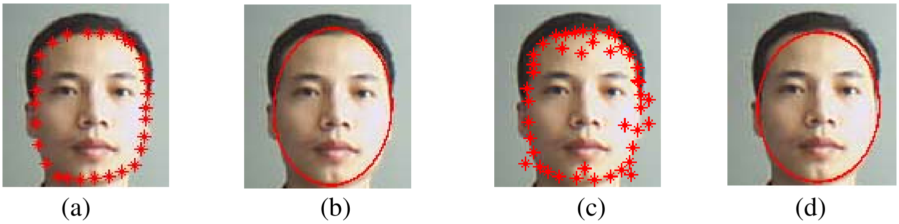

(a) (b) (c) (d)

Fig. 4. The demonstration of direct least squares ellipse fitting for face. (a) The points denoted by hand. (b) The results of direct least squares ellipse fitting corresponding to Fig.(a). (c) The points with outliers denoted by hand. (d) The results of direct least squares ellipse fitting corresponding to Fig. (c).

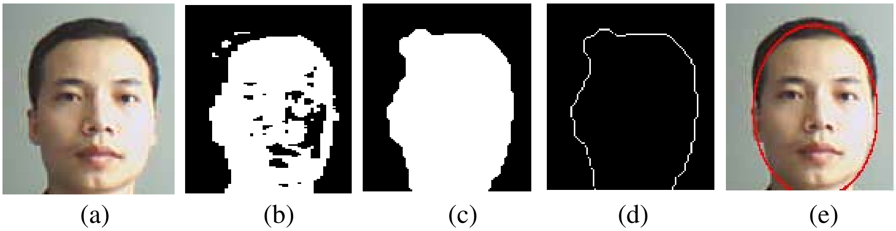

(a) (b) (c) (d) (e)

Fig. 5. The realization process of direct least squares ellipse fitting for face. (a) A target region. (b) The skin region. (c) The result of morphological image processing. (d) The face profile. (e) The result of direct least squares ellipse fitting for face.

The above ellipse fitting algorithm is simple and efficient, furthermore, it is insensitive to the outliers (noise). So we choose this algorithm to describe the elliptical face shape. Firstly, we test the validity of this algorithm by means of denoting the face profile by hand. Fig.4(a) and (c) show the points of face profile denoted by hand. Fig. 4(b) and (d) give the results of direct least squares ellipse fitting corresponding to Fig. 4 (a) and (c), respectively. From the given results we can see this algorithm can work well even in the case that data samples comprise of outliers just as Fig.4(c).

In this paper, the purpose that we use the direct least squares ellipse fitting algorithm is to adjust the face scale automatically during the mean shift tracking process. The realization method is as follows. Firstly, we get the approximate face location using mean shift tracker. Then we attain the skin region via skin detection in the region determined by mean shift tracking process and fill the holes in skin region by means of morphological image processing. Finally, we extract the profile of skin region and use theses profile points as samples to attain the ellipse facial shape by direct least square ellipse fitting. Fig.5 demonstrates the realization method that uses direct least squares ellipse-fitting algorithm to describe the elliptical face shape.

4 Experimental Results

Our algorithm has been proved to be effective by means of lots of experiments. The experimental condition is as follows.

We choose HSV color space to describe the face target, the reason is HSV color space is based on human perceive, it distinguishes the illumination and chrominance explicitly. In order to reduction the effect of illumination, we use fewer bins for V component when we establish the kernel color histogram, specifically, HSV color space is quantized as $16\times16\times8$ bins.

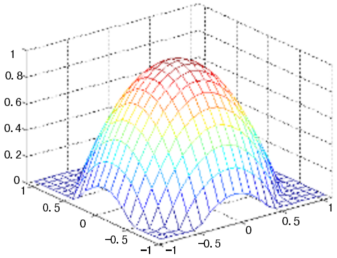


Fig. 6. The Epanechnikov kernel

We choose Epanechnikov kernel for mean shift, the Epanechnikov kernel can be expressed as

$$K_E(\mathrm{x}) = \begin{cases} \dfrac{1}{2} c_d^{-1}(d+2)(1-\|\mathrm{x}\|^2) & \|\mathrm{x}\| \le 1 \\ 0 & \text{others} \end{cases} \tag{9}$$

where c_d is the volume of unit sphere in d dimensional Euclidean space. Fig.6 shows the Epanechnikov kernel, from Fig.6 we can see that Epanechnikov kernel assigns smaller weights to pixels farther from the center when we use it to establish the kernel histogram, the above property of Epanechnikov kernel has the advantage that it can increase the robustness of tracking system because the peripheral pixels are ready to be affected by occlusion or interference from background.

We test our algorithm using lots of video sequences based on the above experimental conditions. At the same time, we give the tracking results using the original mean shift traking algorithm [4] on the same experimental condition. Fig.7 shows the tracking result comparison.

From the tracking result of original mean shift tracker given by Fig. 7 (a) we can see, as for Seq_mb sequence, the part tracking results are not precise because of the local extremums when the face target become smaller and smaller. On the other hand, when the face target become larger and larger, the original mean shift tracker can always give the correct location of the target, but the scale attained is little larger than the true target scale, this can be seen clearly from the tracking results of Seq_sb sequence in Fig. 7 (a).

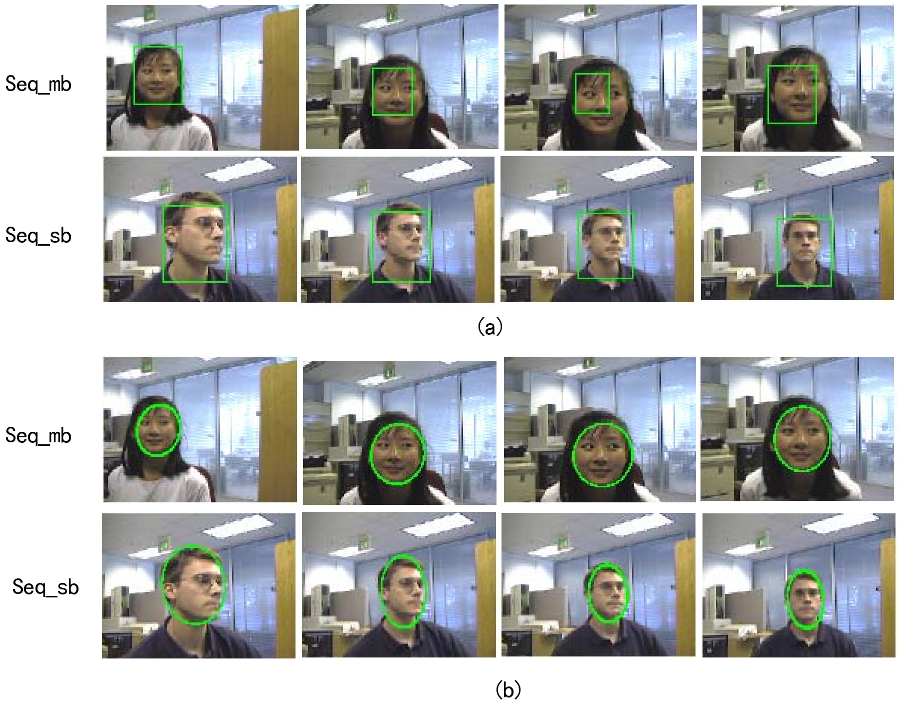


Fig. 7. The tracking results comparison. (a) Results of the original mean shift tracker. (b) Results of the tracking algorithm in this paper.

As to our algorithm, it comprises two steps in fact, one is to locate face target using mean shift tracker with larger kernel bandwidth, and the other is to adjust the face scale automatically using the direct least squares ellipse fitting. Because the two steps can always give the correct face target location and scale respectively, so the two steps complement each other and can track the face effectively. It can be seen clearly from Fig. 7 (b) that our tracking algorithm can deal with the change of target scale well. It can get perfect tracking result whether the targets become larger or smaller gradually. It's clear that the performance of our algorithm is superior to the original mean shift tracking algorithm.

5 Conclusion

The mean shift tracking algorithm is an efficient and simple technique for object tracking. However, it has a shortcoming that it can't adapt scale with object during tracking process. There are presently no effective ways to solve this problem. In accordance with the fact that mean shift tracker always can give the right target location when a larger kernel bandwidth is chosen and the facial shape can be appropriate to an ellipse, this paper introduces a new idea that uses direct least square ellipse fitting to adjust the facial scale based on the mean shift tracking framework. In fact, our algorithm comprises two steps, one is to locate face target using mean shift tracker with a larger kernel bandwidth, and the other is to adjust the face scale automatically using the direct least squares ellipse fitting. Because the two steps can always give the correct face target location and scale respectively, so the two steps complement each other and can track the face effectively. The experimental results demonstrate the efficiency of this algorithm. Its performance has been proven superior to the original mean shift tracking algorithm.

References

1. Nummiaro, K., Koller-Meier, E.,Van Gool, L.: An Adaptive Color-Based Particle Filter. Image and Vision Computing, (2002) 1: 1–12
2. Perez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In European Conference on Computer Vision, (2002) 661–675
3. Fitzgibbon, A., Pilu, M. , Fischer, R.B.: Direct Least Square Fitting of Ellipse. IEEE Transactions on Pattern Analysis and Machine Intelligence, (1999) 21(5): 476–480
4. Comaniciu, D., Meer, P., Ramesh, V.: Kernel-Based Object Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, (2003) 25(5): 564–577
5. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, (2002) 24(5): 603–619
6. Cheng Y.: Mean Shift,Mode Seeking,and Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, (1995) 17(8): 790–799

Improving the Generalization of Fisherface by Training Class Selection Using SOM^2

Jiayan Jiang[1], Liming Zhang[1], and Tetsuo Furukawa[2]

[1] E.E. Dept. Fudan University, 220 Handan Road, Shanghai, China
jiangjiayan@citiz.net, lmzhang@fudan.edu.cn
[2] Kyushu Institute of Technology, Kitakyushu 808-0196, Japan
furukawa@brain.kyutech.ac.jp

Abstract. Fisherface is a popular subspace algorithm used in face recognition, and is commonly believed superior to another technique, Eigenface, due to its attempt to maximize the separability of training classes. However, the obtained discriminating subspace of the training set may not easily extend to unseen classes (thus poor generalization), as in the case of enrollment of new subjects. In this paper, we reduce the performance variance and improve the generalization of Fisherface by automatically selecting some representative classes for training, using a recently proposed neural network architecture SOM^2. The experiments on ORL face database validate the proposed method.

1 Introduction

Face recognition has become an active research topic for decades of years due to its value in both theory and application. To solve this problem, a great number of techniques have been developed, among which Eigenface, a PCA-based algorithm [1], and Fisherface, an LDA-based algorithm [2], are very popular ones.

Although it is argued that LDA may not always outperform PCA, especially when the training samples per class are insufficient or ill-sampled [3], it is a common belief that LDA is superior to PCA, since it tends to maximize the separability of the training classes [2]. However, in most previous work related to LDA, the classes are fixed during the training and testing phases, i.e. the subjects (not the images) being tested are always those involved in training phase.

On the other hand, training is a standalone process prior to the enrollment of subjects in some large-scale face recognition test-beds. The face sample database is usually divided into a development set for training, a gallery which contains the images to be enrolled, and a probe set which comprises of unknown faces to be identified [4, 5]. It should be noted that the training set does not contain all the subjects in the gallery, just as in a real problem. Once training is accomplished, re-training is impractical because it requires updating millions of existed records [4].

It remains unclear whether Fisherface, an LDA-based algorithm especially tuned for training classes, can also perform well on unseen classes in the gallery. This is in fact a generalization problem. This paper aims at improving the generalization of Fisherface by selecting some representative training classes using a recently proposed

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 278 – 285, 2006.

neural network architecture SOM^2 which has been applied in data class visualization and interpolation [6, 7].

The remaining of this paper is arranged as follows: a brief review of Eigenface and FisherFace is given in Section 2; The algorithm of SOM^2 and its application in training class selection of Fisherface are described in Section 3; Section 4 gives experimental results on a publicly available face database, the ORL face database; Finally conclusion is drawn in Section 5.

2 Background

Eigenface is a classical subspace face recognition algorithm proposed in [1]. It is based on the observation that all face samples, which are one-dimension representations of face images, reside in a relatively small subspace, called "face space", compared with the original image space. Thus a classical dimensionality reduction technique PCA is employed to derive such a subspace, which is spanned by the eigenvectors corresponding to the m largest eigenvalues of the sample covariance matrix. These eigenvectors are referred to as Eigenfaces because their appearances are like faces when displayed as images. Although the features derived from Eigenfaces capture most variances of the samples, they are not optimal for classification purposes, for the variances are caused by not only the intrinsic differences of faces (the identities) but also the unwanted extrinsic factors such as lighting conditions.

To overcome the drawback of Eigenface and make use of the label information of the training samples, several LDA-based algorithms are proposed [2], among which Fisherface is the most famous one. Assume that N training samples $\{\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_N\}$ belong to I classes $\{X_1, X_2, \cdots, X_I\}$, the aim of LDA is to select the projection matrix W in order that the ratio of the between-class scatter and the within-class scatter is maximized, i.e.

$$\begin{aligned} W_{LDA} &= \arg\max_{W} \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|} \\ &= \left[\vec{w}_1, \vec{w}_2, \cdots, \vec{w}_m\right] \end{aligned} \tag{1}$$

The between-class scatter matrix is defined as $S_B = \sum_{i=1}^{I} N_i \left(\vec{u}_i - \vec{u}\right)\left(\vec{u}_i - \vec{u}\right)^T$ and the within-class scatter matrix is defined as $S_w = \sum_{i=1}^{I} \sum_{\vec{x}_j \in X_i} \left(\vec{x}_j - \vec{u}_i\right)\left(\vec{x}_j - \vec{u}_i\right)^T$, where N_i is the number of samples in X_i, and $\vec{u}_i$, $\vec{u}$ are the mean vector of the samples in X_i and the grand mean vector of all samples respectively. If S_w is nonsingular, the solution is given by the eigenvectors corresponding to the $(I-1)$ non-zero eigenvalues of $S_w^{-1} S_B$; Otherwise PCA is employed first to make S_w full-ranked. Similarly, if these eigenvectors are treated as images, they are referred to as Fisherfaces.

It can be seen that Eigenface aims at deriving a general face subspace. If the training samples are sufficient, a test face image can also be projected into this subspace effectively, and the classification is performed within it. On the other hand, the attention of Fisherface is mainly focused on deriving a subspace in which the separability is maximized between *training classes*, which generally results in better classification performance than Eiganface with regard to these classes.

However, it is not evident that the separability can be easily extended to unseen classes, as in the case of enrollment of new subjects. Although a conjecture is proposed in [2] that "Fisherface methods, which tend to reduce within-class scatter for all classes, should produce projection directions that are also good for recognizing other faces besides the ones in the training set.", it is not validated by experiments.

We notice that in practice the training classes of Fisherface are usually randomly selected from a large dataset [5]. In the worst case Fisherface may be trained on some "noise" (non-representative) classes, thus the discriminating performance will be poor when confronted with new classes. From a statistical viewpoint, since each time a random training set is used, the variance of performance can be large across different trials. Our idea is that if some representative training classes can be selected from the whole dataset, the performance variance may be reduced and the generalization of Fisherface may be improved.

3 Training Class Selection Using SOM^2

Suppose that a face database includes N samples $\{\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_N\}$ belonging to I classes $\{X_1, X_2, \cdots, X_I\}$, it is more often than not that only a subset of the database can be used in Fisherface training. Our goal is to automatically select some representative (prototype) classes from the whole dataset to form the training set so that the generalization is improved compared with an arbitrary manual selection. Unfortunately, classical techniques of VQ family, such as K-Means, Neural Gas [8], or SOM [9], do not provide us any solutions to this problem, since they only induce some reference (codebook) vectors without any class formation. In our case we need a method which enables density approximation in terms of *classes* rather than *samples*. In this paper, we use SOM^2, a newly proposed neural network architecture, to achieve this end.

SOM^2 is short for "SOM of SOMs" [6, 7] which is a hierarchical structure of self-organizing maps, see Fig.1. Several nodal units (squares with grid) are arranged in array within a parent SOM. Each nodal unit is also a SOM itself called child SOM, which is trained to represent a data manifold. In the mean time, these child SOMs are interacting via the grand parent SOM, which finally generates a self-organizing map representing the distribution of data manifolds. The algorithm of SOM^2 consists of three processes: the competitive process, the cooperative process, and the adaptive process. These processes are iterated until the result is converged or a maximum number of iterations is reached.

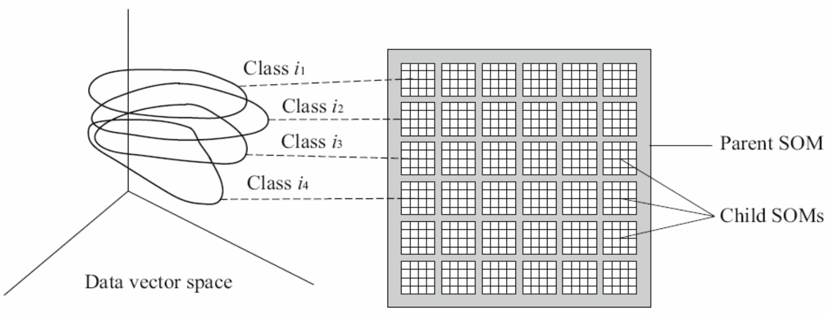


Fig. 1. The scheme and architecture of SOM^2 as "SOM of SOMs"

Suppose that SOM^2 comprises of K child SOMs, each of which has L codebook vectors $W^k = \left\{ \vec{w}^{k,1}, \cdots, \vec{w}^{k,L} \right\} \quad (k = 1, 2, \cdots, K)$. The competitive process includes the competition inside each child SOM and the competition between child SOMs (i.e. in parent SOM). For sample $\vec{x}_j \in X_i$, the competition inside the k th child SOM is to determine the "best matching unit (BMU)" $l^{*k}_{i,j}$, which is defined as follows:

$$l^{*k}_{i,j} = \arg\min_{l} \left\| \vec{w}^{k,l} - \vec{x}_j \right\|^2 \quad \left(\vec{x}_j \in X_i \right) \tag{2}$$

The average error of the k th child SOM for all samples in class i is calculated as:

$$e_i^k = \frac{1}{N_i} \sum_{j=1}^{N_i} \left\| \vec{x}_j - \vec{w}^{k, l^{*k}_{i,j}} \right\|^2 \quad \left(\vec{x}_j \in X_i \right) \tag{3}$$

where N_i is the number of samples in class i. The competition between child SOMs is to determine the "best matching map (BMM)" for class i, based on the average error e_i^k:

$$k^*_i = \arg\min_{k} e_i^k \tag{4}$$

It is obvious that each class can find its corresponding BMM, i.e. the child SOM which minimizes the average error.

In the cooperative process, the learning rates for parent SOM and child SOMs are calculated. The normalized learning rate of the k th child SOM for class i is:

$$\phi_i^k = \frac{g\left[d\left(k, k^*_i\right), T \right]}{\sum_{i'=1}^{I} g\left[d\left(k, k^*_{i'}\right), T \right]} \tag{5}$$

And the normalized learning rate of the l th codebook vector for $\vec{x}_j \in X_i$ is:

$$\varphi^{l}_{i,j} = \frac{h\left[d\left(l,l^{**}_{i,j}\right),T\right]}{\sum_{j'=1}^{N_i} h\left[d\left(l,l^{**}_{i,j'}\right),T\right]} \tag{6}$$

Here $d(\cdot,\cdot)$ refers to the distance between two nodes in the map space, $g[\cdot,\cdot]$ and $h[\cdot,\cdot]$ are the neighborhood functions of parent and child SOMs respectively, whose amplitudes decrease monotonically with increasing d . The neighborhoods also shrink with iteration T . $l^{**}_{i,j}$ denotes the BMU in the BMM for $\vec{x}_j \in X_i$, i.e. $l^{**}_{i,j} \triangleq l^{*k^*_i}_{i,j}$.

In the adaptive process, all codebook vectors of all child SOMs are updated as follows:

$$\vec{w}^{k,l} = \sum_{i=1}^{I} \phi^{k}_{i} \sum_{\vec{x}_j \in X_i} \varphi^{l}_{i,j} \vec{x}_j \quad l = 1,\cdots,L; k = 1,\cdots,K \tag{7}$$

In our setting, SOM^2 is working in the “class density approximation” mode, i.e. the number of child SOMs is smaller than that of the data classes, or $K < I$. It can be regarded as an analogy of conventional SOM in the “point density approximation” mode: as the codebook vectors of a conventional SOM are “representative samples” (prototypes) of the training samples, the child SOMs of a SOM^2 form “representative classes” of the training classes. Inside each child SOM, it leaves flexible whether to approximate or to interpolate the data distribution, depending on the number of codebook vectors per child SOM and the number of samples per class. Like conventional SOM, topology is preserved, not only in child SOMs but also in parent SOM. What’s more, the child SOMs are aligned in the sense that all codebook vectors with the same index share some similar attributes.

All of the $K \times L$ codebook vectors of SOM^2 are used in Fisherface training. Since these training classes are more representative than those randomly selected ones, they can be helpful to reduce the performance variance and improve the generalization of Fisherface, which will be validated in the next section.

4 Experiments

The ORL face database contains different images of 40 subjects, with 10 images per subject. These images includes variations of lighting conditions, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the subjects are in the upright, fontal position, with tolerance for some side movement. The images are grayscale with a resolution of 92×112. Ten images of one subject of the ORL database are shown in Fig. 2. No preprocess is involved in the experiment.

Fig. 2. Ten images of one subject from ORL face database

Please note that we are interested in the case of insufficient training classes and investigating the generalization of Fisherface. Thus we first divide the whole database into two partitions: a candidate training set which includes the first 5 images of all subjects, and a test set including the rest 5 images of all subjects. The candidate training set is used to train SOM^2, thus $I = 40$ and $N_i = 5 \quad (i = 1, \cdots, 40)$. We assign $K(< I)$ child SOMs, each one comprised of 5 codebook vectors, i.e. $L = 5$. In this experiment, both the parent SOM and the child SOMs are one-dimensional maps, and the neighborhood functions are Gaussian, whose standard deviations shrink exponentially with iterations. After 1000 iterations, these child SOMs are regarded as some representative classes for Fisherface training, and the codebook vectors within them are samples belonging to different training classes.

For comparison, K classes are randomly selected from the candidate training set for Fisherface and Eigenface training. Then the whole candidate training set serves as a gallery so that all the images in it are enrolled into the trained recognition system. At last, a Nearest Neighbor classifier is applied to determine the identity of each image in the test set based on the cos-similarity between a test image and the enrolled class centers. For each choice of K, 20 trials are conducted to determine the mean and standard deviation of recognition error rates. Fig. 3 plots the mean error rates and standard deviations with respect to feature dimensions when $K = 20$. The results of different K s are listed in Table 1 and visualized in Fig. 4, where the feature dimensions are fixed at $(K - 1)$ and $(K \times L - 1)$ for Fisherface and Eigenface respectively.

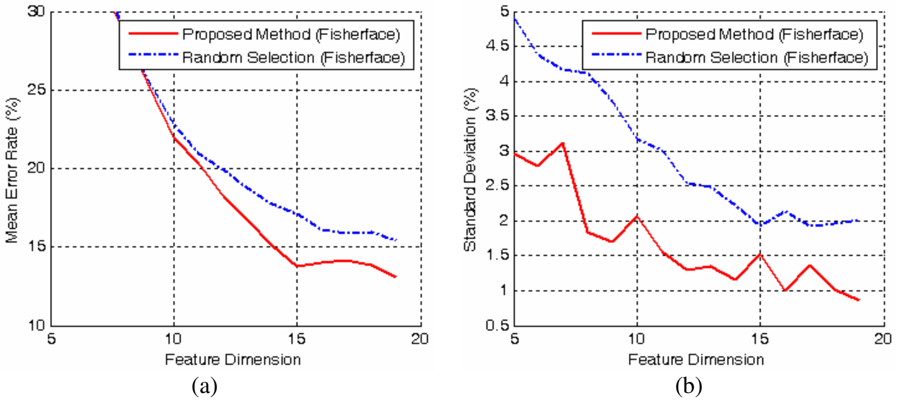


Fig. 3. Mean error rates and standard deviations w.r.t. feature dimensions over 20 trials when $K = 20$. (a) Mean error rates; (b) Standard deviations.

Table 1. Mean error rates and standard deviations for different K s over 20 trials

K	**Proposed Method (Fisherface)**	Random Selection (Fisherface)	Random Selection (Eigenface)
10	**24.10%±2.52%**	27.30%±3.38%	25.00%±1.43%
15	**18.43%±1.09%**	19.03%±2.74%	20.40%±1.40%
20	**13.10%±0.87%**	15.43%±2.01%	18.25%±1.45%
25	**10.20%±1.34%**	12.53%±2.85%	17.13%±0.86%
30	**11.35%±0.69%**	11.73%±1.67%	16.82%±0.94%
35	**8.97%±1.04%**	11.43%±1.82%	16.00%±0.71%

It can be seen clearly that the proposed method improves the performance of Fisherface trained from random selection, either in the sense of mean error rates or standard variations. To summarize, the following discoveries can be obtained: 1) When the training classes are extremely insufficient ($K = 10$), Fisherface is inferior to Eigenface, and then it outperforms Eigenface when more classes are involved in training. This phenomenon is quite similar to that in [3], but the cause in [3] is insufficient data per class for training, rather than insufficient classes in our case; 2) The performance variances of Eigenface are always smaller than those of Fisherface, although the mean error rates are higher. This can be explained from a generalization perspective: since Eigenface is more successful in deriving a general face representation with the training samples, it is more statistically stable than Fisherface; 3) The proposed method effectively improves the generalization of Fisherface, which results in lower error rates, and reduces the performance variances, which are comparable to those of Eigenface.

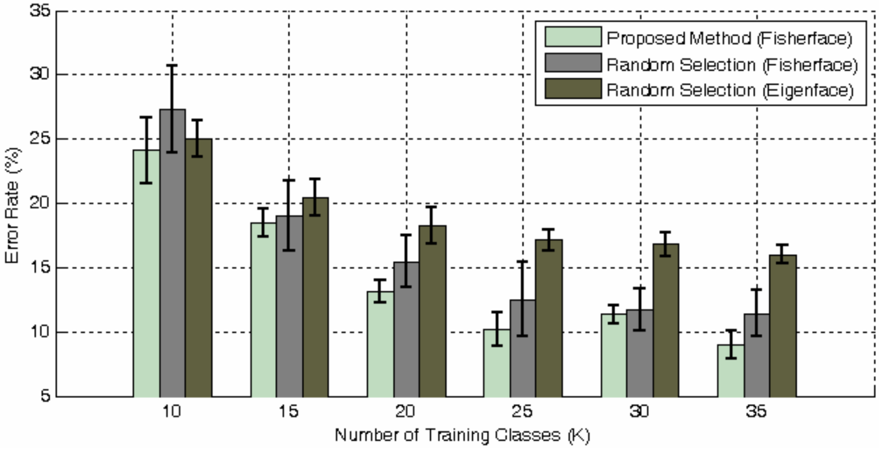


Fig. 4. Error rates w.r.t. number of training classes (K). The bars denote the mean error rates, and the lines denote the standard deviations over 20 trials.

5 Conclusion

Although there have been a great number of papers published in the face recognition area, few of them investigate the impact of training set. A good start point is in [10], where some statistical properties of PCA (Eigenface) are studied. Along this line, we focus on the generalization problem of Fisherface in this paper. We first remind that the optimal discriminating subspace of the training set may not easily extend to unseen classes, as in the case of enrollment of new subjects; then we propose a method to reduce the performance variance and improve the generalization of Fisherface by selecting some representative training classes using a recently proposed neural network architecture SOM^2. The experiments on ORL face database validate this method. In the future, some larger-scale face databases, such as FERET or CAS-PEAL-R1, will be used to investigate the statistical behavior of Fisherface.

Acknowledgement

The authors would like to thank AT&T Laboratories Cambridge for providing the ORL face database. This work was supported by NSFC 60571052 and Significant Technology Project in Shanghai 045115020.

References

1. Matthew A.T. and Alex P.P.: Face Recognition Using Eigenfaces. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (1991) 586-591
2. Peter N.B., Joao P.H., and David J.K.: Eigenface vs. Fisherface: Recognition Using Class Specific Linear Projection. IEEE Trans. on Pattern Anal. Machine Intell. Vol. 19 (1997) 711-720
3. Aleix M.M. and Avinash C.K.: PCA versus LDA. IEEE Trans. on Pattern Anal. Machine Intell. Vol. 23 (2001) 228-233
4. Phillips P.J., Hyeonjoon Moon, Rizvi S.A., and Rauss P.J.: The FERET Evaluation Methodology for Face-Recognition Algorithms. IEEE Trans. on Pattern Anal. Machine Intell. Vol. 22 (2000) 1090-1104
5. Bo C., Shiguang S., Xiaohua Z., and Wen G.: BaseLine Evaluation on the CAS-PEAL-R1 Face Database. Lecture Notes in Computer Science (LNCS3338), Advances in Biometric Person Authentication (2004) 370-378
6. Tetsuo F.: SOM of SOMs: Self-organizing Map Which Maps a Group of Self-organizing Maps. Proc. ICANN (2005) 391-396
7. Tetsuo F.: SOM^2 As “SOM of SOMs”. Proc. WSOM (2005)
8. Martinetz T.M., Berkovich S.G., Schulten K.J.: “Neural-Gas” Network for Vector Quantization and its Application to Time-Series Prediction. IEEE Trans. on Neural Networks, Vol. 4 (1993) 558-569
9. Kohonen T.: Self-Organizing Maps, 3rd.ed., Springer (2001)
10. Phillips P.J., Flynn P.J., Scruggs T., et al.: Overview of the Face Recognition Grand Challenge. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2005) 947-954

Image Registration with Regularized Neural Network

Anbang Xu and Ping Guo[*]

Image Processing Pattern Recognition Laboratory
Beijing Normal University, Beijing, 100875, P.R. China
abxu@ieee.org, pguo@ieee.org

Abstract. In this paper, we propose a new method to improve the image registration accuracy in feedforward neural networks (FNN) based scheme. In the proposed method, Bayesian regularization is applied to improve the generalization capability of the FNN. The features extracted from the image sets by kernel independent component analysis (KICA) technique are input vectors of regularized FNN. The outputs of the neural network are those translation, rotation and scaling parameters with respect to reference and observed image sets. Comparative experiments are performed between FNN with regularization and without regularization under various conditions. The results show that the proposed method is much improved not only at accuracy but also remarkably at robust to noise.

1 Introduction

Image registration is the process of aligning two or more images of the same scene. Image registration techniques are embedded in a lot of visual intelligent systems, such as robotics, target recognition, remote medical treatment and autonomous navigation. The common image registration methods are divided into two types: intensity-based methods and feature-based methods. The analysis and evaluation for various techniques and methods of image registration are carried out on the basis of these two sorts, while the feature-based methods are emphasized.

Recently, Itamar Ethanany [1] proposed to use feedforward neural network (FNN) to register a distorted image through 144 Discrete Cosine Transform (DCT)-base band coefficients as the feature vector. But this method has too large lumber of input feature vectors for the un-orthogonality of DCT based space, thus suffered low computational efficiency and high requirements on computer performance. Later, Wu and Xie [2] used low order Zernike moments instead of DCT coefficients to register affine transform parameters but the estimation accuracy is still not satisfied. We proposed to use the complete isometric mapping (Isomap)[3] and kernel independent component analysis (KICA) [4] for feature extraction. Although the performance of investigated feature extraction methods is better, the generalization of FNN needs to be improved further.

[*] Corresponding author.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 286–293, 2006.

Main challenge in FNN applications is that over fitting problem happens when a neural network over learnt during the training period. As the result, such a over-trained FNN may not perform well on unseen data set due to lack of generalization ability. A good generalized FNN can be obtained with model selection or regularization techniques [5]. Because that the computational efficiency of some model selection methods such as cross validation and bootstrap is very low, in this paper we intend to adopt regularization technique for generalization. We use KICA for feature extraction and then these features are fed into a FNN which is trained by using Bayesian regularization to obtain register affine transform parameters. Experimental results show that the scheme we proposed is better than other methods in terms of accuracy and robustness.

This paper is organized as follows: In section 2, the KICA and regularized FNN based image registration scheme and its algorithm are presented. Section 3 focuses on experimental results comparison with the other methods under different neural network structures and various noisy conditions. Finally, the conclusions are presented in section 4.

2 KICA and Regularized FNN Based Image Registration Scheme

The image registration scheme consists of two stages: the pre-registration phase and the registration phase. In the pre-registration phase, first, a training set is synthesized by the reference image. The feature coefficients are extracted from the training set with the method of KICA, and then these feature coefficients as inputs are fed to a FNN. Second, a neural network is trained with regularization and its target outputs are affine parameters. In the registration phase, since the neural network is trained, the remainder work is simple: We just use the same method to extract features from the registered image and feed these features to the trained network to get the estimated affine parameters.

The background and algorithms are briefly introduced as following sections.

2.1 Affine Transformation

Geometrical transformation can be represented in many different ways, affine transformation is one of the most common used transformations. An affine transformation is the transformation that preserves collinearity. Geometric contraction, expansion, dilation, reflection, rotation, shear, similarity transformations, spiral similarities, translation as well as these combinations are all belonging to affine transformations,. In this paper, we adopt the affine transformation which is the composition of rotations, translations, dilations. Images can be represented with two dimensional matrices and the affine transformation can be described by the following matrix equation [1]:

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + s \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} . \quad (1)$$

In the equation, there are four basic parameters for the transformation, where (x_1,y_1) denotes the original image coordinate, (x_2,y_2) denotes the transformed image coordinate in another image, t_x, t_y are the translation parameters, s is a scaling factor and θ is a rotation angle. In this paper, we will adopt this transformation model.

2.2 Kernel Independent Component Analysis

KICA is a nonlinear method that has been used widely to perform data redundancy reduction and feature extraction. Recently, Liu and Cheng *et al* proposed a new algorithm that incorporates ICA and the kernel trick to improve face recognition [6] and texture classification [7].

The main idea of KICA is to map the input data into an implicit feature space F firstly: $\mathbf{\Phi} : x \in \mathbf{R}^N \to \mathbf{\Phi}(x) \in F$.Then ICA is performed in F to produce a set of nonlinear features of input data.The input data X is whitened in feature space F. The whitening matrix is:$\tilde{\mathbf{W}}_{\Phi} = (\mathbf{\Lambda}_{\Phi})^{\frac{-1}{2}}(\mathbf{V}_{\Phi})^T$, here $\mathbf{\Lambda}_{\Phi}$,$\mathbf{V}_{\Phi}$ are the eigenvalues matrix and eigenvectors matrix of covariance matrix $\hat{\mathbf{C}}=\frac{1}{n}\sum_{i=1}^{n} \mathbf{\Phi}(x_i)\mathbf{\Phi}$ $(x_i)^T$, respectively. Then we can obtain the whitened data $\mathbf{X}_{\Phi}^{W}$ as

$$\mathbf{X}_{\Phi}^{W} = (\tilde{\mathbf{W}}_{\Phi})^T\mathbf{\Phi}(x) = (\mathbf{\Lambda}_{\Phi})^{-1}\alpha^T\mathbf{K}, \tag{2}$$

where K is defined by $K_{ij} := (\mathbf{\Phi}(x_i)\mathbf{\Phi}(x_i))$ and α is the eigenvectors matrix of K. After the whitening transformation, Then $\mathbf{W}_{\Phi}$ can be calculated by the following iterative algorithm:

$$\tilde{\mathbf{Y}}_{\Phi} = \mathbf{W}_{\Phi}\mathbf{X}_{\Phi}, \tag{3}$$

$$\Delta\mathbf{W}_{\Phi} = [\mathbf{I} + (\mathbf{I} - \frac{2}{1+e^{-\tilde{\mathbf{Y}}_{\Phi}}})]\mathbf{W}_{\Phi}, \tag{4}$$

$$\tilde{\mathbf{W}}_{\Phi} = \mathbf{W}_{\Phi} + \rho\Delta\mathbf{W}_{\Phi} \to \mathbf{W}_{\Phi}, \tag{5}$$

until $\mathbf{W}_{\Phi}$ converged, and ρ is a learning constant. According to the above algorithm, the feature of a test data s can be obtained by:

$$y = \mathbf{W}_{\Phi}(\mathbf{\Lambda}_{\Phi})^{-1}\alpha^T\mathbf{K}(\mathbf{X}, s), \tag{6}$$

where $\mathbf{K}(\mathbf{X}, s) = [k(x_1, s), k(x_2, s), ...k(x_n, s)]^T$, k is a kernel function.

In the above iteration algorithm, the function $\mathbf{\Phi}$ is an implicit form. The kernel function k can be computed to instead of $\mathbf{\Phi}$. This trick is named as Kernel Trick. Many functions can be chosen for the kernel such as polynomial kernel:

$$k(x, s) = (x \cdot s)^d \tag{7}$$

Gaussian kernel $k(x, s) = exp(- \parallel x - s \parallel^2 /2\sigma^2)$ and sigmoid kernel $k(x, s) = tanh(k(x{\cdot}s){+}\Theta)$. Liu and Cheng *et al* use a cosine kernel function [6], [7] derived from the polynomial kernel function as shown in Eq.(7), which can give a better performance than the polynomial kernel function for feature extraction:

$$\tilde{k}(x, s) = \frac{k(x, s)}{\sqrt{k(x, x)k(s, s)}}, \tag{8}$$

where k is a polynomial kernel. In previous work, we proved that the performance of the KICA is better than DCT, Zernike, Isomap ans KPCA [4]. In this paper, we still adopt cosine kernel ICA in our experiments.

2.3 Image Registration with Regularized FNN

The image registration scheme includes training the FNN to provide the required affine parameters. Each image in the training set is generated by applying an affine transformation. The affine parameters are randomly changed in a predefined range so as to reduce correlations among images. In order to improve the generalization and immunity of the FNN from over-sensitivity to distorted inputs, we introduce noise in the image synthesis. Then we employ KICA as a feature extraction mechanism presented to the FNN.

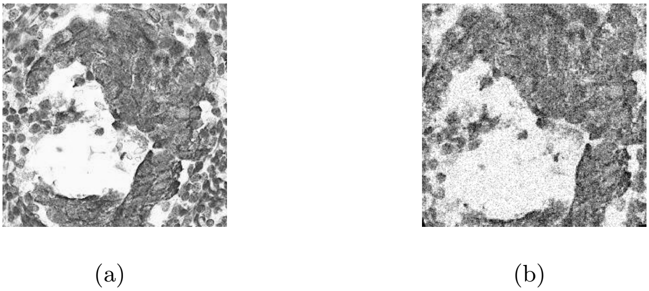

(a) (b)

Fig. 1. (a).An original image and (b).a registered image in the training set with 13 degree rotation, 120% scaling, translation of -2 pixel and 3 pixel on X-axis and Y-axis respectively at a signal-to-noise ratio (SNR) of 15 dB

A good generalized FNN can be obtained with Bayesian regularization. This involves modifying the objective function, which is normally chosen to be the sum of squares of the network errors on the training set. The typical performance function that is used for training FNN is the mean sum of squares of the network errors (MSE).

$$J = MSE = \frac{1}{N_s}\sum_{i=1}^{N_s} \|z_i - g_i\|^2, \tag{9}$$

where N_w is the number of samples in the training set, z_i and g_i is the target and output vector respectively. In this regularization technique, the mean of the sum of squares of the network weights (MSW) is also considered:

$$MSW = \frac{1}{N_w}\sum_{i=1}^{N_w} w_i^2, \tag{10}$$

where N_w represents the number of network weight parameters and w_i an element of the matrix in a vector expression $\mathbf{W}$. The modified objective function is

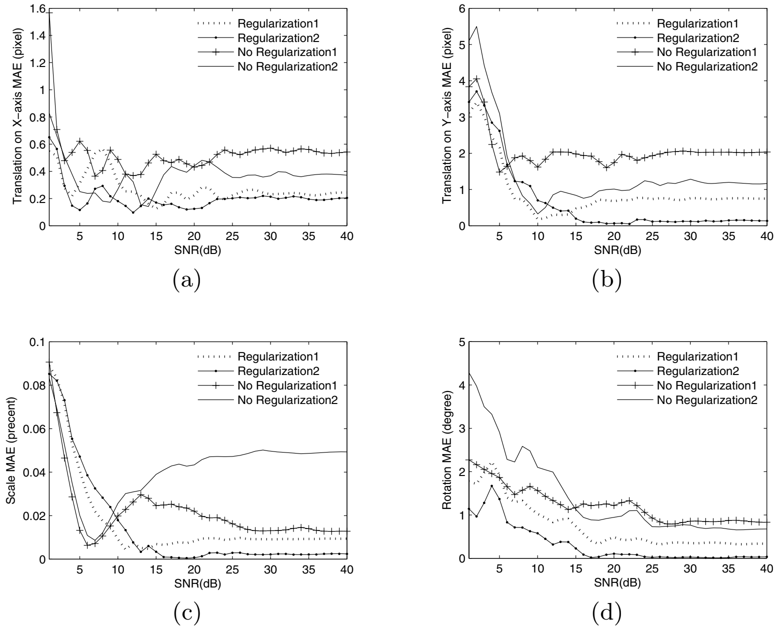
$$J = \beta MSE + \alpha MSW, \tag{11}$$

where α and β are the regularization parameters which are to be optimized in Bayesian framework of MacKay [8], [9]. In minimizing this objective function to find the network weight parameter, the effective value of the regularization parameter depends only on the dimension of weight parameter vector. The weights and biases of the network are assumed to be random variables and follow Gaussian distributions. It is a known fact that the optimal regularization technique requires quite costly computation of the Hessian matrix. To overcome this drawback, Gauss-Newton approximation to the Hessian matrix is used. The approximation with Levenberg-Marquardt algorithm for network training is adopted in this paper [10], [11], [12], [13]. Here the structure of the FNN is that contains 60 inputs, 4 outputs. Sigmoid transfer functions are employed in the hidden layers while linear functions characterize the output-level neurons.

3 KICA and Regularized FNN Based Image Registration Scheme

3.1 Different Number of Hidden Neurons

In the experiment, we compare the performance of FNN with regularization and without regularization under different number of hidden neurons and various



(a)

(b)

(c)

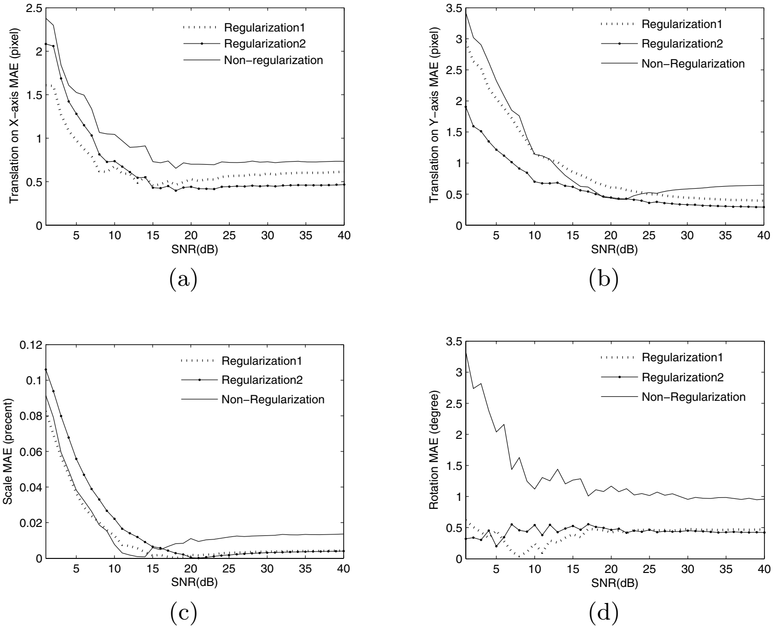
(d)

Fig. 2. Absolute registration error comparison under different number of hidden neurons and noisy conditions. (a) error with respect to translation on X-axis, (b) translation on Y-axis, (c) scale and (d) rotation errors.

noisy conditions. A pair of 256×256 resolution images was used. Fig. 1 shows one of the original images and a transformed image by the translation, rotation and scaling.

The training set consists of 300 images, each image is transformed from the reference image by translating, rotating and scaling randomly within a predefined range. Besides, additive Gaussian noise and Salt & Pepper type noise are applied on each image in various intensities. We also generate some test samples to demonstrate the registration accuracy of the proposed method. We apply KICA to the training samples and reduce the dimension of the sequence of vectors from 65536 to 60. These feature coefficients of images are inputs of FNN, the FNN is trained with Bayesian regularization and its outputs are affine parameters. Finally, the feature coefficients are extracted from the registered image with the same method and fed as inputs into the trained neural network to get the estimated affine parameters.

In order to evaluate registration performance with Gaussian noise, we take 40 images for each the evaluated SNR value. The test image is rotated 13 degree, 120%scaled, translated -2 pixels and 3 pixels on X-axis and Y-axis respectively, as shown in Fig. 1(a). Fig. 2 depicts the results of estimating the affine transform parameters under different SNR values by using the method with regularization and without regularization. The number of hidden neurons in Regularization1



(a)

(b)

(c)

(d)

Fig. 3. Absolute registration error comparison under different number of training samples and noisy conditions. (a) error with respect to translation on X-axis, (b) translation on Y-axis, (c) scale and (d) rotation errors.

and No Regularization1 is 20. The number of hidden neurons in Regularization2 and No Regularization2 is 30. As can be seen from the results, the performance of regularized FNN is more accurate than the method without regularization especially when the structure of the FNN is complex.

3.2 Different Number of Training Samples

In this experiment, comparisons are made between FNN with regularization and without regularization under different number of training samples and various noisy conditions. Similarly, we use test image "Cameraman" which is rotated 15 degree, scaled 77%, translated -5 pixels and 4 pixels on X-axis and Y-axis respectively. Fig. 3 described the results of estimating the affine transform parameters under different SNR values. The number of training samples in Regularization1 is 100. The number of training samples in Regularization2 and No Regularization is 200. As can be seen from the results, our proposed scheme is better than the method with regularization. Even if the number of training samples is reduced from 200 to 100, the performance of the proposed method still shows more accurate than the method without regularization.

4 Conclusions

In this paper, a new method is proposed to improve the accuracy of image registration, which adopts the regularized FNN and KICA to register affine transform parameters. The regularized FNN performs well in estimating affine parameters, especially as the structure of the neural network is complicated and the number of training samples is small. Experiment results show that the proposed method has more accurate registration performance and robust to noise than some other methods. In the future work, other regularization parameter estimation method also can be exploited to improve generalization abilities of the FNN [14].

Acknowledgement

The research work described in this paper was fully supported by a grant from the National Natural Science Foundation of China (Project No. 60275002).

References

1. Elhanany, I., Sheinfeld, M., Beckl, A., *et al*: Robust Image Registration Based on Feedforward Neural Networks. IEEE International Conference on System, Man and Cybernetics, Vol.2 (2000) 1507–1511
2. Wu, J., Xie, J.: Zernike Moment-based Image Registration Scheme Utilizing Feedforward Neural Networks. The 5th World Congress on Intelligent Control and Automation, Vol.5 (2004) 4046–4048

3. Xu, A.B., Guo, P.: Isomap and Neural Networks based Image Registration Scheme. Lecture Notes in Computer Science, Vol. 3972. Springer- Verlag, Berlin Heidelberg (2006) 486–491
4. Xu, A.B., Jin, X., Guo, P., Bie, R.F.: KICA Feature Extraction in Application to FNN based Image Registration. The 2006 International Joint Conference on Neural Networks, (to appear)
5. Guo, P.: Studies of Model Selection and Regularization for Generalization in Neural Networks with Applications. PhD Thesis, the Chinese University of Hong Kong (2001)
6. Liu, Q.S., Cheng, J., Lu, H., Ma, S.: Modeling Face Appearance with Nonlinear Independent Component Analysis. In: Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR2004), Vol.2 (2004) 761–766
7. Cheng, J., Liu, Q.S., Lu, H.: Texture Classification Using Kernel Independent Component Analysis. In: Proceedings of the 17th Int. Conf. on Pattern Recognition, Vol.1 (2004) 620–623
8. MacKay, D. J. C.: Bayesian interpolation. Neural Computation **4** (3) (1992) 415–447
9. MacKay, D. J. C.: A practical Bayesian framework for backpropagation networks. Neural Computation **4** (3) (1992) 448–472
10. Marquardt, D.: An Algorithm for Least-squares Estimation of Nonlinear Parameters. In: SIAM Journal Applied Mathematics, Vol.2 (1963) 431–441
11. Hagan, M.T., Menhaj, M.: Training Feedforward Networks with Marquardt Algorithm. IEEE Trans. Neural Networks **1** (1) (1994) 113–118
12. Foresee, F. D., Hagan, M. T.: Gauss-Newton approximation to Bayesian regularization. In: Proceedings of the 1997 International Joint Conference on Neural Networks, (1997) 1930–1935
13. Doan, C.D., Liong, S.Y.: Generalization for Multilayer Neural Network: Bayesian Regularization or Early Stopping. In: Proceedings of Asia Pacific Association of Hydrology and Water Resources 2nd Conference (2004)
14. Guo, P., Lyu, M. R., Chen, C. L. P.: Regularization Parameter Estimation for Feedforward Neural Networks. IEEE Trans. Neural Networks **33** (1) (2003) 35–44

A Statistical Approach for Learning Invariants: Application to Image Color Correction and Learning Invariants to Illumination

B. Bascle, O. Bernier, and V. Lemaire

Orange / France Telecom R & D

Abstract. This paper presents a new approach for automatic image color correction, based on statistical learning. The method both parameterizes color independently of illumination and corrects color for changes of illumination. The motivation for using a learning approach is to deal with changes of lighting typical of indoor environments such as home and office. The method is based on learning color invariants using a modified multi-layer perceptron (MLP). The MLP is odd-layered. The middle layer includes two neurons which estimate two color invariants and one input neuron which takes in the luminance desired in output of the MLP. The advantage of the modified MLP over a classical MLP is better performance and the estimation of invariants to illumination. The trained modified MLP can be applied using look-up tables (LUTs), yielding very fast processing. Results illustrate the approach.

1 Introduction

The apparent color of objects in images depends on the color of the light source(s) illuminating the scene. Because of this color constancy problem, image processing algorithms using color, such as color image segmentation or object recognition algorithms, tend to lack robustness to illumination changes. Such changes occur frequently in images (shadows, lights on/off, varying sunlight). To deal with this, a color correction scheme that can compensate for illumination changes is needed.

2 Illumination Correction – State of the Art

Color in images is usually represented by a triband signal, for instance Red-Green-Blue (RGB). As discussed in the introduction, this signal is sensitive to changes in illumination. However, image processing techniques need to be robust to such changes. Therefore color needs to be parameterized independently of illumination. This can be done by parameterizing color with one or two parameters or by correcting the triband signal. A number of color parametrization and color correction schemes have been described in the literature [9]. This section describes a number of approaches that work on a single image. Table 1 summarizes their pros and cons.

Examples of directly correcting the triband signal are diagonal color correction (such as gray world and white patch) and non-diagonal color correction [6]. They are both

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 294–303, 2006.

Table 1. Comparison of color correction approaches that work on a single image

approach	principle of the approach	local / global	cons	pros
estimation of illuminant color [1]	neural network estimates illuminant chromaticity from image uv histogram	global	same illuminant for whole image, further processing for image correction	illuminant explicitly identified
ratio-based color invariants [2]	analytic color invariants	local / pixel-wise	original image can't be reconstructed from invariant images	fast
luminance correction in HSV space [3]	simple analytic color correction	local / pixel-wise	completely local, relatively sensitive to illumination changes	very fast using LUTs
color transfer [4]	normalization by mean and variance in $l\alpha\beta$ color space	global	limited to global changes in illumination	fast
intrinsic image by entropy minimization [5]	finds an axis invariant to illuminant color by entropy minimization, then projects image perpendicularly to axis	global	need for few colors and many illuminations in image to find invariant axis, not fast	works for any illuminants
diagonal color correction	linear	global	restricting assumptions, no non-linearities	very fast
non-diagonal color correction [6]	PCA-based linear correction	pixel	illuminants must be known	fast (using LUTs)
enhancement of dark images using modified multi-scale retinex [7]	multi-scale convolution (linear)	local areas	color correction for visual effect, performance for background subtraction unknown	fairly fast (3 fps for 640x480 images), any lighting (blueish, etc ...)
color correction using a ”classic” MLP [8]	statistical learning of non-linear color correction transform by MLP	pixel with learnt global a priori for lighting	trained for given rear projection setup & lighting conditions, does not estimate color invariants	could be very fast (using LUTs)
color correction using a trained modified MLP (this paper)	statistical learning of non-linear color correction transform by MLP + statistical learning of 2 color invariants	pixel with learnt global a priori about type of lighting	trained for range of lightings (e.g. customary in home and office e.g. whitish or yellowish)	very fast (LUTs, 3.75 ms per frame or 266 fps for 320x240 images), trained for range of illuminations

linear, and cannot model non-linearities. They also rely on limiting assumptions (known image mean for gray world, known maximum value for each channel for white patch, illuminants known for [6]). They are very fast and can be implemented using LUTs for even greater speed.

In [8] a neural network is used to learn the color correction needed in a specific rear projection environment. It does not estimate color invariants. It also is trained for specific lighting conditions.

An example of mono-band parametrization of color is hue (from hue-saturation-value, a.k.a. HSV) [3]. An example of bi-band color parameterization are chrominances uv (from the YUV color space) [3] and the ab values from the CIE Lab color space [3]. These three color representations (H, uv or ab) are analytical and thus do not require learning. They are fast pixel-wise methods. They have a certain robustness to illumination changes, but this robustness is limited. Color transfer [4] is a method with a similar philosophy, normalizing color by its mean and variance in $l\alpha\beta$ space. It is global and fast, but limited to global changes in illumination.

An approach for estimating color invariants from images consists in calculating ratios of RGB components at a given pixel (R/B) or between neighboring pixels (such as $(R_{x_1}G_{x_2})/(G_{x_1}R_{x_2})$) [2]. This method is also pixel-wise and thus fast. These invariants are also very robust to illumination changes. However, a lot of information about the original signal is lost and reconstructing it from the invariants is difficult.

A more sophisticated method has been proposed by [5]. It estimates a mono-band invariant and is based on a physical model of image formation. It works globally from the whole image. In $(log(R/B), log(G/B))$ color space, an axis invariant to illuminant color is determined by entropy minimisation. Projecting the image perpendicularly to the axis gives corrected colors. The approach does not require learning and applies to any type of illuminant, but is relatively slow. It also requires that the image contains relatively few different colors and many changes of illumination for each color.

Yet another approach consists in explicitly estimating the color of the illuminant [1]. A neural network estimates the chromaticity of the illuminant from the histogram of chromaticity of the whole image. The method works globally from the whole image and supposes there is only one illuminant for the entire image.

Another method is [7]. It is a bit out of the scope of this paper, since it aims at the enhancement of dark images for visual effect, and does not give information about performance for color correction. However, it gives a benchmark about speed, since the authors aimed at fast processing. This will be discussed in section 4.5.

3 A Statistical Approach to Measure Color Invariants

3.1 A Modified Multi-layer Perceptron: Motivation

The motivation of this work is twofold: (1) to parameterize color compactly and independently of illumination by two invariants (2) to do it in real-time. Firstly, two parameters are needed to parameterize color with enough degrees of freedom to reconstruct a triband signal, given a luminance (or a gray level signal). Secondly, real-time processing (25/30 images per second for video) is also necessary for some applications. For this, slow methods such as [1] and [5] are unsuitable. Pixel-wise approaches are more suited. Among those, hue-Saturation, uv (from YUV) and ab (from the CIE Lab color space) lack robustness to illuminations changes. [2] is robust to these, but reconstructing an image from the invariant(s) is difficult. A new fast approach is needed.

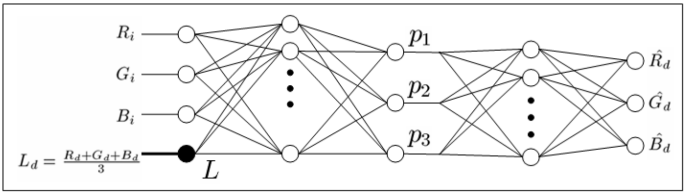


Fig. 1. A classical MLP with 4 inputs can be used to perform color correction. (R_i, G_i, B_i) is the input color. (R_d, G_d, B_d) is the desired output color, corresponding to the same color seen under a different illumination. $L_d = \frac{R_d+G_d+B_d}{3}$ is the luminance of the desired output and is a direct function of the illumination.

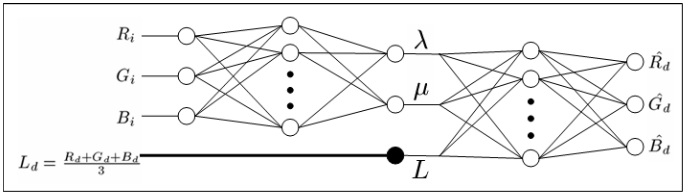


Fig. 2. A modified MLP for color correction and color invariant learning. λ and μ are the color parameters invariant to illumination that the MLP is trained to estimate. $(\hat{R}_d, \hat{G}_d, \hat{B}_d)$ are the actual outputs of the network. Bias neurons are omitted from this figure.

In practice, a limited range of illuminants are available in indoor environments. It is therefore interesting to use learning methods to find a color parameterization invariant to the ”usual” illumination changes. This also provides a priori information about the illuminants, making the color correction global, which is, as Land showed [10]), necessary to perform correct illuminant correction. In practice, the lighting usually found in home and offices comes from fluorescent lights, incandescent light bulbs and natural sunlight from windows. They tend towards the whitish and yellowish areas of the spectrum (very few bluish or reddish lights). These are the illuminants that our approach deals with.

Our learning method of choice is neural networks and more specifically multi-layer perceptrons (MLPs) for their ease of use and adaptability. A classic MLP with 4 input neurons and 3 output neurons can be used for color correction under varying illuminations (see fig. 1). The fourth input, a context input, is the luminance L of the expected output and is a direct function of the illumination. This fourth input neuron prevents the mapping to be learnt by the MLP from including one-to-many correspondences (the different corrected colors corresponding to the same input color with different illuminations) and thus makes it solvable. If the MLP contains a bottleneck layer with 3 neurons, then these perform a re-parameterization of RGB space. However the three color parameters estimated by the 3 neurons (called here $p_1p_2p_3$) have no reason to be invariant to illumination.

To force the MLP to code color independently of illumination, the architecture of the traditional MLP is modified (see fig. 2). The entry point L of the MLP (fourth input neuron) is moved to the bottleneck layer of the network so that it becomes the third and last neuron of this layer. This displaced entry makes our MLP different from a trivial compression network. The two other neurons of the bottleneck layer have outputs (λ, μ). During training, the network learns to reconstruct the corrected color (R_d, G_d, B_d) from (λ, μ) and the desired output luminance $L_d = \frac{R_d+G_d+B_d}{3}$. Thus it learns to ignore the luminance of the input (R_i, G_i, B_i) and learns to estimate two color characteristics (λ, μ) that are invariant to illumination.

The approach does not require any camera calibration or knowledge about the image. However, it supposes that the illuminants are of the type commonly found in indoor environments.

3.2 Training the Modified Multi-layer Perceptron

As shown in fig. 2, the modified MLP includes 5 layers (this could be generalized to an odd number of layers). The input and output layers have 3 neurons each (plus an additional bias), for RGB inputs and outputs. The middle layer includes 3 neurons (excluding bias): their outputs are called λ, μ and L. The second and fourth layers have arbitrary numbers of neurons (typically between 3 and 10 in our experiments). The links between neurons are associated to weights. Neurons have sigmoid activation functions. The network includes biases and moments [11].

A database of images showing the same scenes under different illuminations is used to train the modified MLP. The illuminations are typical of indoor environments such as home and office.

A classic MLP training scheme based on backpropagation is applied. A pixel is randomly sampled at each iteration from the training set. Its RGB values before and after an illumination change (from real images) are used as input (R_i, G_i, B_i) and desired output (R_d, G_d, B_d) to the network. Propagation and back-propagation are then performed, with one modification: as mentioned above, the output L of the third neuron of the third layer is forced to the value of the luminance corresponding to the desired output color.

3.3 Use of the Modified Multi-layer Perceptron

The trained modified MLP can be used to correct color images. Each image pixel is propagated through the first half of the trained network to find the invariants λ and μ. An arbitrary luminance L is imposed on the pixel by forcing the output of the third neuron of the third layer to L. The output of the trained network then gives the corrected color. If a constant luminance L is used for all pixels in the image, an image corrected for shadows and for variations of illumination across the image and between images is obtained. The color correction can be tabulated for fast implementation. The approach could be easily extended to a greater number of inputs and outputs or different inputs/outputs than RGB. For instance, YUV or HSV, or redundant characteristics such as RGBYUVLab could be used as inputs and outputs.

4 Image Correction Results

4.1 Experimental Conditions and Database

The network was trained using 546000 pixels, randomly sampled from 91 training images (6000 pixels per image), taken by 2 webcams (Philips ToUCam Pro Camera and Logitech QuickCam Zoom). The training images are of indoor scenes viewed under different illuminations typical of home and office environments. Testing was performed on other images taken by the 2 webcams used for training and by a third webcam, not used for training, a Logitech QuickCam for Notebooks Pro.

In practice, using 8 neurons in the second and fourth layers of the MLP gives good performance. A gain of 1.0 was used, with a momentum factor of 0.01 and a learning rate of 0.001. Pixels that were too dark (luminance ≤ 20) or too bright / saturated (luminance ≥ 250) were not used for training.

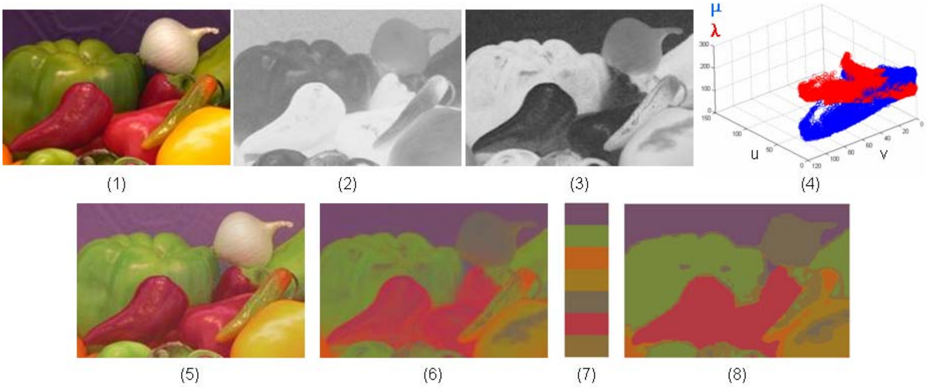


Fig. 3. Example of color correction learnt by the modified MLP. (1) original image (unknown illumination). (2) and (3) invariants λ and μ estimated by the MLP. (4) locus of the invariants in the uv space. (5) corrected image with pixel luminance inputs set to values proportional to pixel luminances in the original image (plus a constant). (6) corrected image with the pixel luminance inputs set to a constant value for all pixels. (7) 7 color peaks found by mean shift [12] in the corrected image (6). (8) resulting image segmentation.

4.2 Comparison with a ”classical” Multi-layer Perceptron

Table 2 shows that the modified MLP (fig. 2) performs better in reconstructing target images than a classic MLP (fig. 1). The reconstruction is done given the expected luminances L_d of the pixels of the desired target image.

4.3 Invariant Estimation by the Modified MLP

Figure 3 shows the two invariants (λ, μ) learnt by the modified MLP and calculated on an image (see part (1) of fig 3) of unknown illumination. The two invariants are seen in

Table 2. Mean error between reconstructed and target images for a "classical" MLP and the modified MLP presented in this article. The mean error was calculated using 748 320x240 test images (not in the training set). The error is averaged over the three color components (R,G,B).

	for a classical MLP	for the modified MLP
mean error (in pixel values $\in [0, 255]$)	10.47	5.54
relative mean error	4.11%	2.17 %

parts (2) and (3) of the figure. Objects of similar color to the human eye have similar values of λ and μ. Part (4) of fig. 3 shows the locus of the invariant values (λ, μ) in the image as a fonction of the chrominance values (u, v) (from YUV color space) of the image pixels. The locii of the two invariants are not identical, and thus we have two invariants and not only one. Part (6) of figure 3 shows the corrected image estimated for a constant luminance input over the image. Much of the influence of shading and variations of illumination across the image is removed, apart from specularities (white saturated areas) which are mapped to gray by the network. Areas of similar color in the original image (despite shading and illumination) have much more homogeneous color in the corrected image. This is further shown by performing mean-shift based color segmentation [12] on the corrected image. Seven areas of uniform color are readily identified and segmented (see part (7) and (8) of fig. 3) in the corrected image. They correspond roughly to what is expected by a human observer. This example illustrates that our modified MLP successfully learns a parameterization of color by two parameters that are invariant to illumination.

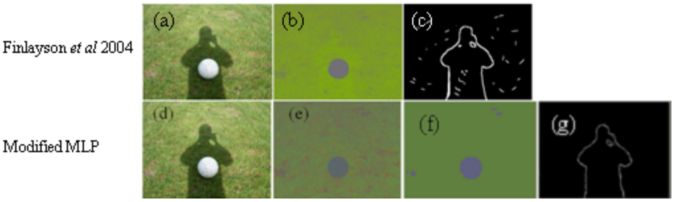


Fig. 4. Comparison of the pixel-wise color correction by the modified MLP presented in this paper and the whole-image color correction method of Finlayson *et al* [5]. Application to shadow detection. Example I. (a) and (d) show the original image. (b) is the invariant image obtained using the method of [5] and (c) shows the shadow edges estimated from (b). (e) shows the corrected image estimated using the modified MLP, (f) and (g) the results of mean shift color segmentation [12] from (e) and (h) the shadow edges estimated from (g).

4.4 Comparison with Other Color Correction Methods from the Literature

Figures 6, 4 and 5 compare our color correction approach with other color correction approaches.

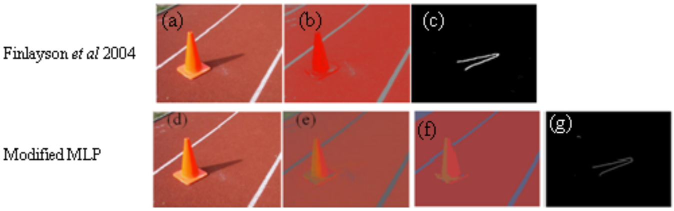


Fig. 5. Comparison of the pixel-wise color correction by the modified MLP presented in this paper and the whole-image color correction method of Finlayson *et al* [5]. Application to shadow detection. Example II. (a), (b), (c), (d), (e), (f), (g) and (h) illustrate the same steps as in fig. 4.

Figure 6 compares our approach to HSV-based color correction and applies it to color-based background subtraction. The two first images of the first and second columns of the figure show that our color correction scheme is indeed robust to changes in illumination, since there is much less difference between the images after correction than before. Figure 6 also shows that the correction performed in this paper compares favorably with an HSV-based color correction (which consists in taking an RGB color to hue-saturation-value space, setting its value/luminance to a constant, then going back to RGB space to get the corrected color).

Figures 4 and 5 illustrate that our correction is of similar quality to that of Finlayson et al [5] (briefly described in the introduction of this paper). The application of color correction is the detection of shadow contours (which can be used for shadow removal, as shown in [5]). Even though it might be less robust to large light changes or unusual light changes (such as turning on a blue or red light), our method is faster, being pixel-wise.

4.5 Performance of a LUT Implementation of the Trained Modified MLP

Color correction by the modified MLP can be tabulated, making it one the fastest possible color correction approaches. Execution time using LUTs is 3.75 ms for an entire 320x240 image, on a Pentium4 3GHz. This way, color correction can be used as a first step in video-rate image processing, without using a large part of the frame processing time (40ms). This LUT implementation is possible because the approach is pixel-wise.

An HSV correction scheme could be as fast (using LUTs), but it would be less performant, as illustrated by fig. 6. A color correction scheme based on [5] would be of equal performance, as illustrated on examples by fig. 4 and 5. It could deal with more changes of illumination, since our approach is limited to the type of frequently found indoor lighting the modified MLP was trained for. However, working globally on the image, it could not be implemented as a LUT, and would thus be slower. The approach of [7] (briefly described in section 2), which performs good-quality color enhancement at good speed, is slower than our approach (3 frames per second on a Pentium4 2.26GHz for a 640x480 image).

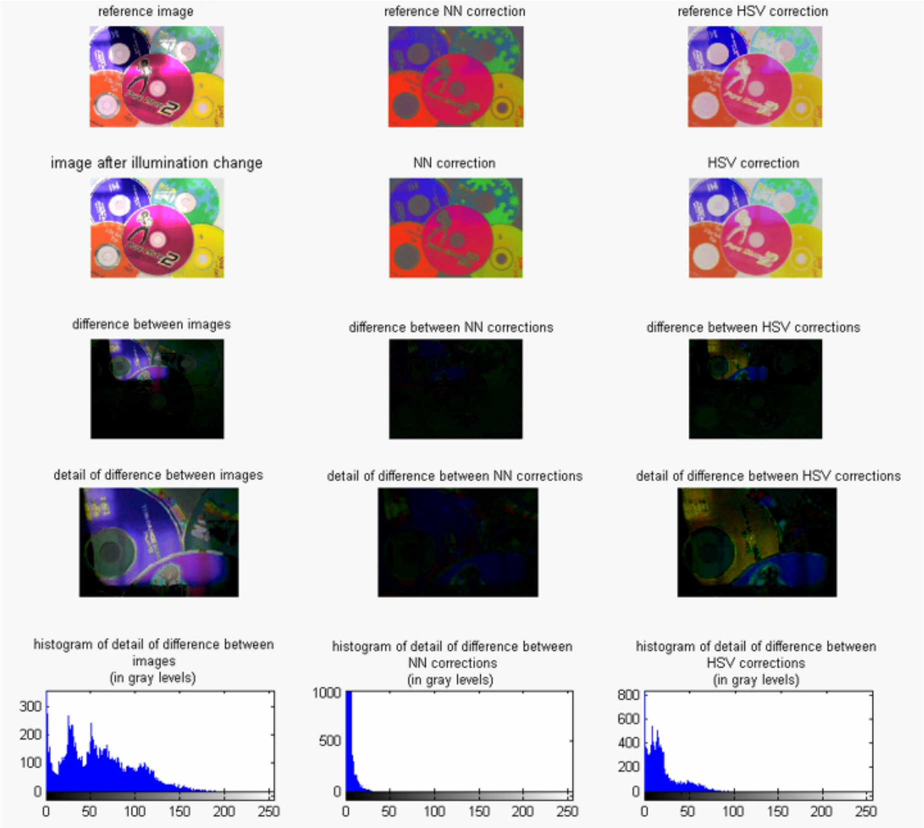


Fig. 6. Comparison of the pixel-wise color correction by the modified MLP presented in this paper and pixel-wise HSV-based color correction, HSV being the well known hue-saturation-value color space

5 Conclusion

This paper presents a new neural network-based approach to estimating image color independently of illumination. A modified multi-layer perceptron is trained to estimate two color invariants and an illumination- corrected color for each input color. The network is trained for typical indoor home and office lighting (fluorescents and light bulbs) and outdoor natural light, using two webcams. Such statistical training gives the approach a good compromise between generality (being able to handle different types of illuminants) and discrimination power (being able to discriminate between different colors). Experiments with lighting changes and another webcam show that the training seems to have good generalization properties. Once learning has been achieved, color correction is very fast using look-up tables, so that color correction can be performed as a part of image pre-processing before applying other image processing algorithms (such as background subtraction or color-based image segmentation).

References

1. Funt, B., Cardei, V., Barnard, K.: Neural network colour constancy and specularly reflecting surfaces. In: Proc. of AIC Color 97, Kyoto, Japan. (1997)
2. Gevers, T., Smeulders, A.: Color based object recognition. Pattern Recognition (1999)
3. Gonzalez, R., Woods, R.: Digital Image Processing. Addison-Wesley Longman Publishing Co., Inc. (2001)
4. Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. IEEE Computer Graphics and Applications (2001)
5. Finlayson, G., Drew, M., Lu, C.: Intrinsic images by entropy minimization. In: Proc. 8th European Conf. on Computer Vision (ECCV'04), Prague, pp 582-595. (2004)
6. Funt, B., Jiang, H.: Non-diagonal colour correction. Proc. International Conference on Image Processing (ICIP 2003), Barcelona (2003)
7. Tao, L., Asari, V.: Modified luminance based msr for fast and efficient image enhancement. Proc. of 32nd Applied Imagery Pattern Recognition Workshop (AIPR'03) (2003)
8. Yin, J., Cooperstock, J.: Color correction methods with applications to digital projection environments. Journal of the Winter School of Computer Graphics (2004)
9. Barnard, K., Martin, L., Coath, A., Funt, B.: A comparison of computational color constancy algorithms, part 2; experiments with images. IEEE Transactions on Image Processing (2002)
10. Land, E., McCann, J.: Lightness and retinex theory. J. Opt. Soc. Am. (1971)
11. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press (1996)
12. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Proc. of IEEE Conf. on Comp. Vis. and Pat. Rec. (CVPR 2000). (2000)
13. Tappen, M., Freeman, W., Adelson, E.: Recovering intrinsic images from a single image. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. (2005)
14. Luo, Q., Khoshgoftaar, T.: Efficient image segmentation by mean shift clustering and MDL-guided region merging. In: Proc. of 16th IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI'04). (2004)
15. Gu, I.H., Gui, V.: Colour image segmentation using adaptive mean shift filters. In: Proc. of . Int. Conference on Image Processing (ICIP'01). (2001)
16. Tominaga, S.: Color coordinate conversion via neural networks. Proc. of International Conf. on Colour Imaging in Multimedia (CIM'98) (1998)
17. Vrhel, M., Trussell, H.: Color scanner calibration via a neural network. Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 99) (1999)
18. Rosenberg, C., Minka, T., Ladsariya, A.: Bayesian color constancy with non-gaussian models (NIPS 2003)

Limited Recurrent Neural Network for Superresolution Image Reconstruction

Yan Zhang, Qing Xu, Tao Wang, and Lei Sun

Zhengzhou Institute of Surveying and Mapping, No. 66 Longhai Middle Road, Zhengzhou
450052, China
zhangyanxz7806@163.com

Abstract. The paper proposes a new method for image resolution enhancement from multiple images using the limited recurrent neural network (LRNN) approach, which is a set of collectively operating feed-forward neural networks. In the limited recurrent networks, information about past outputs is fed back through recurrent connections of output units and mixed with the input nodes flowing into the network input as external input nodes. Thus, experience about past search is utilized, which enables LRNN to be capable of both learning and searching the optimal solution for optimization problems in the solution space. Estimates computed from a low-resolution (LR) simulation image sequence and an actual video film sequence show dramatic visual and quantitative improvements over bilinear interpolation, and equivalent performance to that of the frequency domain approach.

1 Introduction

There are increasing demands for high-resolution (HR) images in various applications, including health diagnosis and monitoring, military surveillance, and terrain mapping by remote sensing. Although the most direct way to increase spatial resolution is to use a HR image acquisition system, the high cost for high precision optics and image sensors is always a prohibitive factor in many commercial applications. Therefore, a new approach toward increasing spatial resolution is required to overcome these limitations of the sensors and optics manufacturing technologies. Currently one most promising approach is to use image superresolution (SR) reconstruction technique to obtain a HR image (or sequence) from the multiple observed LR images[1].

Since Tsai and Huang's work[2], many work has been reported in the literature, including the weighted least-squares algorithm[1], the nonuniform interpolation approach[1], the projection onto convex sets(POCS) method[3] and MAP Bayesian approach[4]. All of these approaches are based on certain assumptions about a degradation imaging model and the statistics of the additive noise. When the degradation factors in the imaging process are ambiguous, the performances of these approaches are limited. To overcome these limitations and reduce computational complexity in SR image reconstruction problem, we try the neural network approach in the paper. Abiss[5] proposed a modified Hopfield neural network for SR image reconstruction, Zhang[6] presented a scheme combining intra-frame interpolation and linear restoration

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 304 – 313, 2006.

by neural network together for image restoration. Wang[7] employed the standard radial basis function(RBF) to realize the functional mapping from the degraded image space to the original image space. Salari[8] proposed the integrated recurrent neural network for image resolution enhancement, combining the Hopfield neural network and the feedforward network together. Hopfield type network is superior in solving optimization problems, but the search for an optimal solution is slow for the experience gained from the prior relaxation is not utilized in the next iteration and the network is reinitialized to the initial state in each relaxation. The multilayered feedforward type networks, as the RBF network, have dominant learning capability, but they lack the capability of searching an optimal solution in the solution space through a relaxation process.

Therefore, in this paper we propose a novel limited recurrent neural network(LRNN) approach for SR image reconstruction from multiple LR images. The proposed LRNN method has the ability of searching an optimal solution in the solution space by relaxation and learning by adjusting the connection weights. The proposed approach is trained by a Gauss-Newton Real Time Recurrent Learning (RTRL) algorithm[9]. Experimental results on a simulation image sequence and an actual satellite image sequence demonstrate that the proposed method is competitive in solving image resolution enhancement problem, for its excellent learning adaptability.

2 Preliminaries on LRNN and SR Image Reconstruction

Let $z(k)=(z_1(k), \ldots, z_N(k))$ be the network state vector (vector of output nodes) at time k, let W be the weight matrix. f denote the node function, such as sigmoid, tanh and etc. The recurrent network dynamic system is defined as,

$$z(k+1)=f[Wz(k)]. \tag{1}$$

where k denotes the time index. In the above definition, all external input nodes and network input nodes are lumped into one vector $z(k)$ for simplicity. The network input nodes are clamped together at the particular input values: $x_i(k) = u_i(k)\ i \in I$, where $u(k)$ represents the network input vector at time k, and I represents the set of network input nodes.

Recurrent networks are used in situations when we have current information to give the network, but the sequence of inputs is also important. We need the neural network to store a record of the prior inputs and combine them with the current data to produce an answer. Fully recurrent networks provide two-way connections between all processors in the neural network. Due to their complex and dynamical property, they exhibit instability and chaotic behavior associated with their power, and take an in-determinate amount of time in reaching a stable state. However, LRNN is a good compromise between the simplicity of a feed-forward network and the complexity of a fully recurrent neural network, allowing feedback from the hidden units or the output units to flow back into the network as a set of external inputs, while prohibiting two-way connections between all nodes.

In training LRNN, the following error function is defined

$$E = \frac{1}{2}\sum_{k=1}^{K}\sum_{i\in O}\left(e_{ki}\right)^2 = \frac{1}{2}\sum_{k=1}^{K}\sum_{i\in O}\left(z_i\left(k\right) - d_i\left(k\right)\right)^2 . \tag{2}$$

where e_{ki} is the residual error of output unit i at time at time k, $d_i(k)$ is the desired output for unit i at time k , and O is the set of output nodes. In the SR image reconstruction, d_i (k) represents unit i in the original HR image and z_i (k) is unit i in the reconstructed image at time k.

In SR image reconstruction, the ill-posed reconstruction problem can be regularized through adding a priori information constraints and its general solution can be expressed as,

$$\hat{\mathrm{z}} = \arg\min\left[\sum_{i=P}\left\|\mathbf{y}_i - \mathbf{H}_i\mathbf{z}\right\|^2 + \alpha\sum_{c\in S}\varphi_c(\mathbf{z})\right]. \tag{3}$$

where P is the number of LR image frames, $\mathbf{y}_i$ is the LR image, $\mathbf{H}_i$ represents the contribution of pixels in the HR image to the corresponding pixels in the LR image, $\mathbf{z}$ is the reconstructed HR image, α is the regularization parameter, $\varphi_c\left(\mathrm{z}\right)$ is a potential function that depends only on the pixel values located within clique c, and S denotes the set of cliques. The first part calculates the difference between the pixel value of the LR images and the downsampled versions of the HR image $\mathbf{z}$. The second part is the regularization component resulting a smooth solution.

3 Proposed Limited Recurrent Neural Network Structure

The LRNN iteratively feeds its output back to the input until it converges from an initial state to a stable state. It carries the advantages of both the recurrent network as well as the multilayered feedforward network in solving optimization problems. The designed LRNN structure includes of three layers, the input layer, the hidden layer and the output layer. Fig. 1 shows the overall diagram of the proposed network. The hidden layer is functionally divided into two parts: the comparison part and the constraint part. The comparison network computes the difference value corresponding to the first part of equation 3. The constraint network functions as an estimator of the second part of equation 3, representing the regularizing constraint information. Finally, the output network combines the outputs from these two networks to obtain

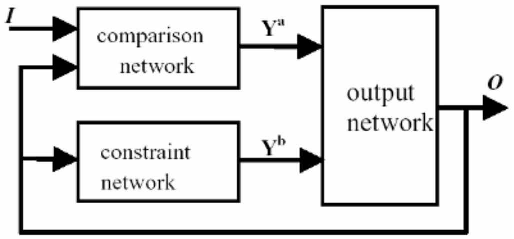


Fig. 1. Diagram of the proposed LRNN for SR image reconstruction

the expected HR image. The details of these networks are described in the following subsections, respectively.

3.1 Comparison Network

Considering the *p*th LR image frame $\mathbf{y}_p$, p=(1, . . .,P), let I_{pk} (p=1,2,. . ., P; k =1, 2, . . ., 9) denote the *k*th pixel value of a 3×3 window $\mathbf{y}_p$, θ_p, $\boldsymbol{h}_p$ =(h_{px} , h_{py}) be the rotation and translation parameters of $\mathbf{y}_p$ relative to the reference LR image. The input nodes vector of the three-layered comparison network is $\boldsymbol{I}$ = (I_{11}, I_{12}, . . ., I_{19}, I_{21}, I_{22}, . . ., I_{29}, . . . , I_{P1}, I_{P2}, . . ., I_{P9}, θ_1, h_{1x}, h_{1y}, θ_2, h_{2x}, h_{2y}, . . ., θ_P, h_{Px}, h_{Py}). The external input vector to the network is the feedback from the output units $O=\left(O_1^n,O_2^n,\ldots,O_M^n\right)$, O_k^n denotes the output unit k in the *n*th relaxation, M is the number of neurons in the output nodes set.

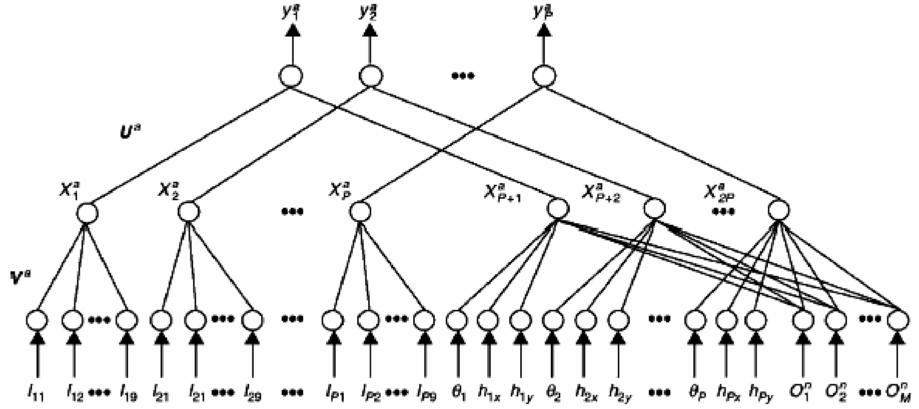

y_1^a
y_2^a
y_P^a
U^a
X_1^a
X_2^a
X_P^a
X_{P+1}^a
X_{P+2}^a
X_{2P}^a
V^a
I_{11} I_{12} ••• I_{19} I_{21} I_{21} ••• I_{29} ••• I_{P1} I_{P2} ••• I_{P9} θ_1 h_{1x} h_{1y} θ_2 h_{2x} h_{2y} ••• θ_P h_{Px} h_{Py} O_1^n O_2^n ••• O_M^n

Fig. 2. Comparison network

In the hidden layer, the first P neurons take the pixel values in 3×3 window in the LR image frames as inputs and compute the mean value of the 9 pixels. Each of the last P neurons calculates the downsampled LR image pixel value based on the information provided by the HR image pixels $O=\left(O_1^n,O_2^n,\ldots,O_M^n\right)$ in the previous relaxation and the related rotational and translational parameters θ_p, $\boldsymbol{h}_p$. Each neuron in the output layer calculates the difference between the *j*th and (P+j) th output neurons in the hidden layer. The implementation of the network is defined as,

$$x_j^a=\sum_{i=1}^{9}I_{ji}\Big/9\ ,\quad j=1,2,\ldots,P\,. \tag{4}$$

$$x_j^a=\theta_{j-P}+h_{(j-P)x}+h_{(j-P)y}+\sum_{i=1}^{M}O_i^n\Big/M\ ,\quad j=P+1,P+2,\ldots,2P\cdot \tag{5}$$

$$y_k^a=x_k^a-x_{P+k}^a\ ,\quad k=1,2,\ldots,P\,. \tag{6}$$

where x_k^a and y_k^a are respectively the output neuron k in the hidden layer and the output unit k in the output layer.

3.2 Constraint Network

The constraint network is designed to be a two-layered feedforward network shown in Fig. 3. The outputs of the network are formulated as follows,

$$y_k^b = \sum_{i=1}^{M} \alpha_{ki} O_i^n \,, \quad k = 1,2,\ldots,M \,, i = 1,2,\ldots,M \,. \tag{7}$$

where α_{ki} is the connection weight among the neurons (Fig. 4) and α_{ki} is set as,

$$\alpha_{ki} = \begin{cases} 1, & \text{for } \; k = i \\ -1/4, & \text{for } \; j : O_k \text{ is a cardinal neighbour of } O_i \end{cases} \,. \tag{8}$$

The constraint network calculates the Laplacian value of each point in the external input, which is fed back from the output of the whole LRNN network. The output of the constraint network is used to control the smoothness of the reconstructed HR image.

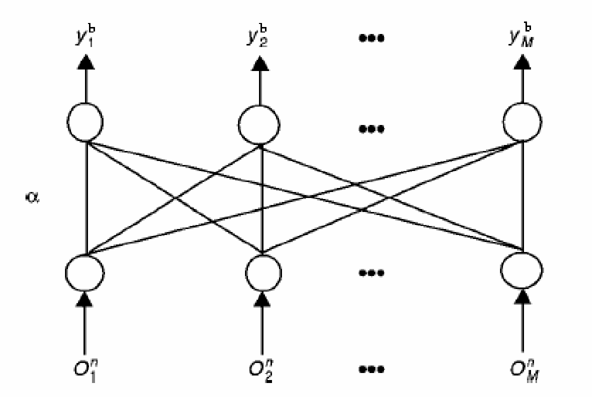


Fig. 3. Constraint network

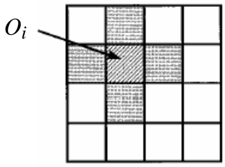


Fig. 4. The cardinal neighbors of a HR pixel O_i. In this case, α_{ki} would be nonzero only O_k is an immediate spatial neighbor of O_i (shaded pixels).

3.3 Output Network

The output network is also a two-layered feedforward network as shown in Fig. 5. It combines the outputs of the two networks in the hidden layer as input vector to generate a set of new outputs. The output can be calculated by the following equation,

$$O_m^{n+1} = f\left(\sum_{k=1}^{P} w_{km} y_k^a + w_{(P+1)m} y_m^b \right). \tag{9}$$

where w_{km} (k =1, 2, . . .,P+1) is the connecting weight between the kth neuron in the input vector and the mth neuron in the output vector.

The activation function is chosen to be the sigmoid function

$$f(x) = 1/\left(1+e^{-x}\right). \tag{10}$$

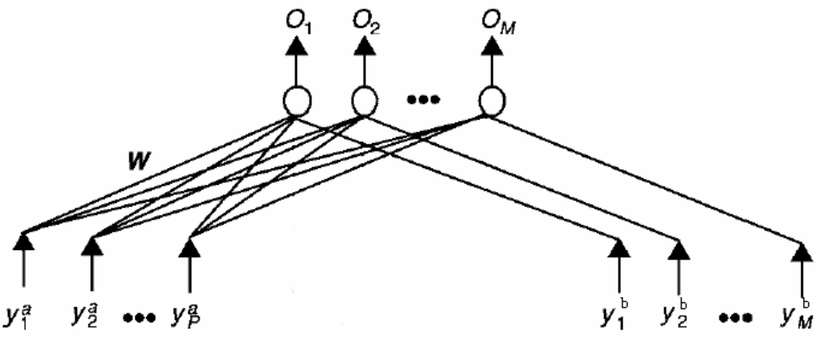


Fig. 5. Structure of Output network

3.4 Training the LRNN Network

We adopt the Gauss-Newton Real Time Recurrent Learning (RTRL) method to train the proposed network due to its simplicity and fast convergence. It allows the network to iteratively search along the negative Gauss-Newton gradient direction. The network is trained by updating weights along the direction vector $\mathbf{p}^{GN}$,

$$w_{km}(\text{ new}) = w_{km}(\text{ old}) - \eta \mathbf{p}^{GN} . \tag{11}$$

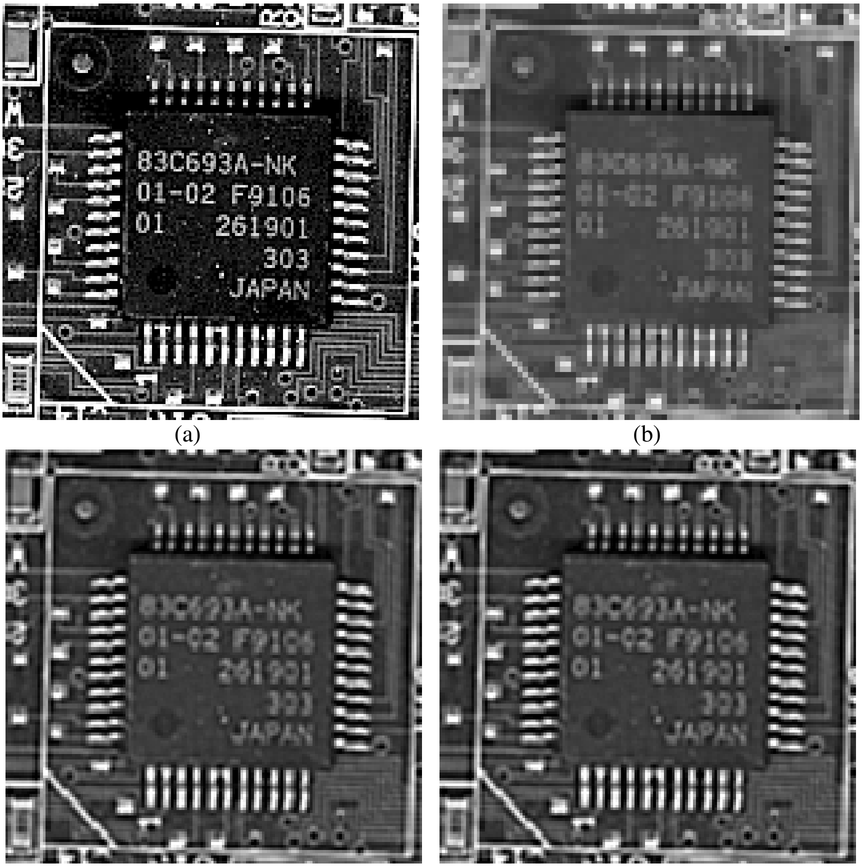
$$\mathbf{p}^{GN} = -\sum_i \frac{u_i^T r(t)}{\sigma_i} v_i \cdot \tag{12}$$

$$r(t) = \left[e_{t0}, e_{t1}, e_{tM}, \ldots, e_{(t-1)0}, \ldots, e_{(t-K+1)M} \right]^T . \tag{13}$$

where η is the learning rate, $r(t)$ is the residual errors of outputs across K consecutive time sequence. $\boldsymbol{u}_i$ is the left SVD decomposition of the Jacobian matrix of the error function E in equation 2, $\boldsymbol{v}_i$ is the right decomposition, σ_i is singular values.

4 Results

The experiment results on the proposed neural network are presented here. The training images set comprises the original HR image and the degraded LR images generated from the HR image. Nine sets of motion parameters (θ_p, $\boldsymbol{h}_p$) are generated by IDL random function, with subpixel magnitudes for $\boldsymbol{h}_p$; and small values in the range of [0°, 3°] for θ_p. Using these parameters, a sequence of 9 translated images is generated from the 512×512 sized HR image. These 9 images are further blurred by 5×5 Gaussian smoothing filter and decimated by a factor of L_1= L_2=4 to produce 9 LR images of size 64×64. Following the above procedure, two training sets of LR images together the corresponding original HR images are obtained using the 'Lena'


(a)

(b)

(c)

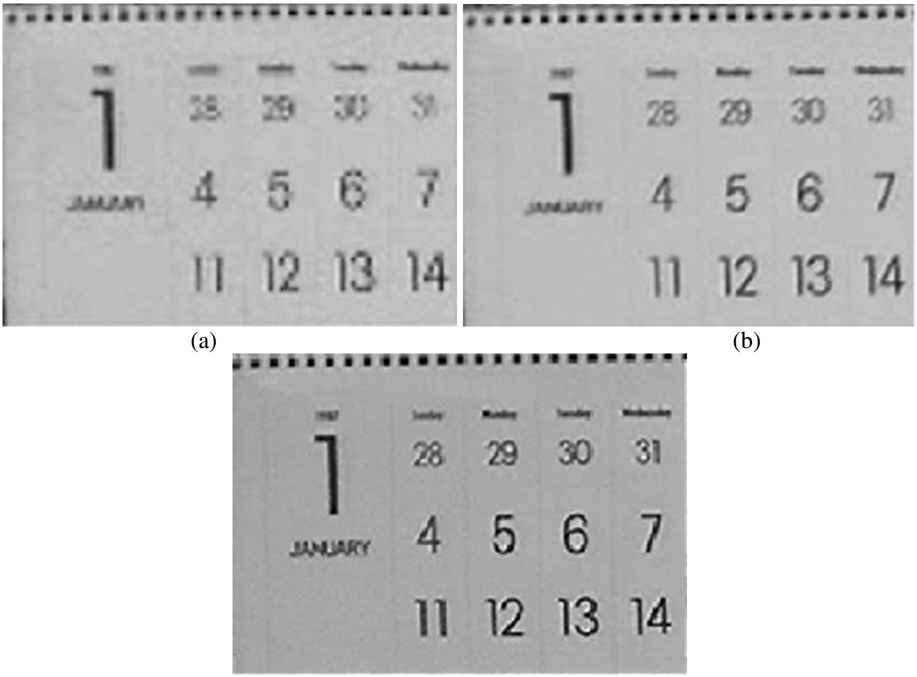
(d)

Fig. 6. LRNN on board sequence. (a) Original HR image. (b) Bilinear interpolation of the reference image. (c)SR reconstruction result by the frequency domain approach. (d) SR reconstruction result by LRNN approach.

and‘Board’ images, respectively. One typical LR image is bilinear interpolated to generate an initial HR image. During the training process, the mean squared error (MSE) corresponding to the original HR image and the current generated HR image is calculated for each iteration. The training process does not terminate until the changes of MSEs between two consecutive iterations reach certain small values.

After the training process, FRNN is used to reconstruct the HR image using the 9 LR ‘Board’ images. The original HR ‘Board’ image is shown in Fig.6 (a). The bilinear interpolated HR image is shown in Fig. 6(b), the HR reconstruction image provided by the frequency domain method[11] is shown in Fig.6(c), and the LRNN reconstructed HR image is shown in Fig. 6(d). The PSNR (Peak Signal-to-Noise Ratio) of the bilinear interpolation is 20.1, that of the frequency domain reconstruction result is 23.1, and that of LRNN result is 23.2. Cleary, the LRNN approach has efficiently improved the spatial resolution of the LR images and its SR reconstruction performance is as excellent as that of the frequency domain approach. The digital numbers, the characters and the circuit nodes in the LRNN result are much clearer than what is seen in the bilinear interpolation result.

To further investigate the performance of the proposed network, we applied LRNN to another set of LR images without retraining. The motion parameters are estimated using a gradient-based motion estimation algorithm [10]. Fig. 7(a), 7(b) and 7(c) show



(a)

(b)

(c)

Fig. 7. LRNN on bridge sequence. SR reconstruction result by (a) bilinear interpolation, (b) the frequency domain approach, (c) LRNN reconstruction.

the generated HR images from bilinear interpolation, the frequency domain reconstruction and LRNN reconstruction, respectively. The visual resolution of the calendar is effectively enhanced in the LRNN reconstruction result.

From the results, it can be seen that the proposed LRNN has the wonderful learning capability to extract the internal mapping relationship between a set of LR image sequence and the HR image. Compared with other SR reconstruction methods in the spatial domain and the frequency domain[1], the proposed LRNN also has the advantage of simplicity and parallel computational capability.

5 Conclusion

We have proposed a novel neural network based method for superresolution image reconstruction from multiple low-resolution image frames. The proposed network is a kind of limited recurrent network which feeds back its outputs to the inputs without any time delay. It has both of the capabilities of learning and searching optimal solutions in the solution space for optimization problems. Simulation results demonstrate that the proposed limited recurrent neural network to SR image reconstruction problems is, therefore, competitive with the traditional methods for solving the problem. We have shown that LRNN can successfully solve the ill-posed high-resolution image reconstruction problem and achieve significant visual improvement. Further, in comparison with other methods, the proposed neural network SR reconstruction is also simple and efficient.

References

1. Park, S. C., Park, M. K., Kang, M. G..: Super-Resolution Image Reconstruction: A Technical Overview. IEEE Signal Processing Magazine. 5 (2003) 21-36
2. Tsai R. Y., Huang, T.S.: Multiframe image restoration and registration. In: in Huang, T.S.(Ed.): Advances in computer vision and image processing, JAI Press, (1984) 317-339
3. Patti, A.J., Sezan, M. I., Tekalp, A. M.: Superresolution Video Reconstruction with Arbitrary Sampling Lattices and Nonzereo Aperture Time. IEEE Trans. Image Processing. 8 (1997) 1064-1997
4. Schulz, R.R., Stevenson, R. L.: Extraction of High-Resolution Frames from Video Sequences. IEEE Trans. Image Processing. 6 (1996) 996-1011
5. Abiss, J.B., Brames, B.J., Fiddy, M.A.,:Superresolution Algorithms for a Modified Hopfield Neural Network. IEEE Trans. Signal Processing. 7(1991) 1516-1523
6. Zhang, L.M., Pan, F.Z.,; A New Method of Image Super-Resolution Restoration by Neural Networks. Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'OZ) , Vol. 5 2414-2418
7. Wang D.H., Talevski, A. Dillon, T.S.,: Edge-Preserving Image Restoration Using Adaptive Components-based Radial Basis Function Neural Networks. 0-7803-7898-9/03, 2003,1243-1248
8. Salari, E., Zhang S.,: Integrated Recurrent Neural Network for Image Resolution Enhancement from Multiple Image Frames. IEEE Proc.-Vis. Image Signal Process, 5(2003).299-305

9. Vartak, A.A., Georgiopoulos, M., Anagnostopoulos G.C.: on-line Gauss-Newton-based learning for Fully Recurrent Neural Networks. Nonlinear Analysis 63(2005) e867-e876
10. Hardie, R.C., Barnard, K.J., Bognar, J.G., Armstrong, E.E., Watson, E.A.: High-resolution Image Reconstruction from a Sequence of Rotated and Translated Frames and its Application to an Infrared Imaging System. Opt.Eng. 1(1998) 247-260
11. Kim, S.P., Bose, N.K., Valenzuela, H.M.: Recursive Reconstruction of High Resolution Image From Noisy Undersampled Multiframes. IEEE Trans. Acoustics. Speech. and Signal Processing, 6(1990).1013-1027

Remote Sensing Image Fusion Based on Adaptive RBF Neural Network

Yun Wen Chen* and Bo Yu Li

Department of Computer Science and Engineering,
School of Information Science and Engineering,
Fudan University, Shanghai 200433, China
{ywchen, liboyu}@fudan.edu.cn

Abstract. With the availability of multi-sensor and multi-frequency image data from operational observation satellites, the fusion of image data has become an important tool in remote sensing image evaluation and segmentation. This paper presents a novel Radius Basis Function (RBF) neural network with some distinctive training strategies, which can integrate multiple information sources efficiently and exploit the potential advantages of each feature. Multi-scale features extracted from remote sensing images are evaluated adaptively and used for segmentation. Experimental results obtained on artificial and real data are both presented which demonstrate the effectiveness of our proposal.

1 Introduction

Digital image fusion is a relatively new research field at the leading edge of available technology. With the availability of multi-sensor and multi-frequency images from Synthetic Aperture Radar (SAR), it forms a rapidly developing area of research in remote sensing [1] [2]. Many image fusion approaches have been developed in literature, including the intensity-hue-saturation transform (IHS) [3], principal component analysis (PCA) [4], discrete wavelet transform (DWT) [5] and so on. However, to properly evaluate various features used in the segmentation process is the main problem remain unsolved. Many fusion algorithms perform on multi-spectral and panchromatic imaging sensors having different ground resolutions of pixels. While many methods use pyramid-based schemes which are complexity and inefficient [1].

Radius Basis Function (RBF) Neural Network is one kind of feed-forward neural network. It has many good advantages, such as simple structure and a good approaching performance. RBF network has been used in various fields including pattern classification, function approaching etc. [9] [15] [14]. However, it is a rather difficult task to train the numerous parameters with the former training strategies and sometimes the RBF approaches give poor performance [16]. When it is applied to remote sensing images the problem will become even severe and desire for a favorable solution.

* Corresponding author.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 314–323, 2006.

Based on these considerations, this paper presents a novel Radius Basis Function (RBF) neural network which adopts the network with feature weights to fuse multi-scale features. We then apply the adaptive RBF neural network to the remote sensing image fusion. A novel multi-phase training strategy is proposed to integrate multiple information sources efficiently. We first train the kernel centers without considering network weight and kernel width, thus we can simplify the training process and locate the centers of hidden unit more accurately because the centers usually play the most important role for classification [17]. Then the rest parameters (the network weights, the feature weights and the kernel variances) are trained simultaneously with fixed center position. During the process some distort effects between parameters are eliminated. As a consequence, improved performance can be obtained. Experimental results obtained on artificial and real data are both given to demonstrate the feasibility of the proposed method. In the final section some discussions and conclusions are presented.

2 Adaptive RBF Neural Network

In order to fuse different features extracted from the SAR image, we use the RBF model with feature weight. Concretely, the kernel function of hidden units is formulated as follows:

$$\phi_j(x) = \exp(\frac{||\mathbf{w}^{L_1}x - \mu_j||}{2\sigma_j^2}) \tag{1}$$

where x represents any pattern vector in the training set and μ, σ are called the center of the kernel function of the $j-th$ hidden unit and its width respectively, $\mathbf{w}^{L_1} = \{w_1, ..., w_m\}$ is the feature weights which is desired to fuse multi-scale features, m reflects the max component of the pattern vector.

As presented in the above section, to tain the RBF neural network is a difficult task because the conventional training scheme is found to be time-consuming and tend to suffer from the local minima problems [9] [6]. Although many complex training technique have been suggested [14] [15] [13] [11], most of all are based on RBF neural network without feature weights. So in this paper, we propose the multi-phase training algorithm based on the error back-propagation process. This method is very novel and effect. Comparing to the clustering method as well as the BP algorithm, we can obtain better generalization performance. The key technique of our method is to locate the centers of hidden units.

In order to locate the center of hidden units accurately, the multi-phase training method divide the hidden units, denoted by $\mathscr{N}_H$, into n_c subset, $\mathscr{N}_H^i(i = 1, ..., n_c)$, according to the proportions of samples of each class in the training set. Then we chose randomly some samples from the λ_i class as the initial center of all neurons of the $\mathscr{N}_H^i$, That is to say, we let the neurons in $\mathscr{N}_H^i$ only serve for the $i-th$ class. Therefore, it becomes possible for us to training the centers respectively using the individual class error, which is the most difference of our method from the others, as a result, we can simplify the training process and eliminate the canceled effect between parameters [16]. In implementing, we initiate the neural network by randomly chose some samples from the each class as the initial center of every neuron in $\mathscr{N}_H^k$. In this paper the vector $\mathbf{w}^{L_2}$ denotes the network weights that connect the hidden layer and the output layer, and they are initialized within $[-1, 1]$. Meanwhile the width δ_j $(j \in [1, |\mathscr{N}_H|])$ are all initialized properly.

2.1 The Multi-phase Training Strategy

Training kernel center and feature weights. Given an input feature vector $\mathbf{x}_i$ with m dimensions, the corresponding value of the $k-th$ output neural is f_k. We mark $d_k = 1$ if $\mathbf{x}_i \in \lambda_k$ (The $k-th$ class) or $d_k = 0$ otherwise. Then the error function can be partitioned into n_c parts denoted as E_k $(k \in \{1, 2, \cdots, n_c\})$.

$$E_k(\mathbf{w}^{L_1}, \mathbf{w}^{L_2}, \mu) = \sum_{\forall \mathbf{x}_i \in \lambda_k} \|f_k(\mathbf{x}_i) - d_k(\mathbf{x}_i)\|_2 \tag{2}$$

The above E_k represents the sum of error corresponding to each class. Consequently, in order to adjust the kernel center of our RBF network, we have

$$\mu_j = \mu_j - \eta \nabla E_k, \quad if\ j \in \mathscr{N}_H^k,\ k \in [1, n_c] \tag{3}$$

where η is a parameter that controls the learning rate. ∇E_k is the gradient of error sum and $j \in \mathscr{N}_H^i$ means the j-th hidden unit is allocated to serve for the k-th class. The above equation indicates that the kernel centers of the RBF network should be adjusted along the gradient-descend direction of the inner-class error. This adjusting method will prevent the center of a hidden unit from escaping far away from its class, which means that it is a local fast adjusting scheme. Furthermore, the feature weight adjusting rules are follows:

$$w_{ji}^{L_1} = w_{ji}^{L_1} - \eta \nabla^2 E_k \tag{4}$$

where $w_{ji}^{L_1}$ is the fusion weight connection the i-th image feature and j-th neural of the hidden layer. $\nabla^2 E_k$ is the second order partial derivative of the k-th class error used here to feedback the adjustment.

Training other parameters of hidden units. Let $E = \sum_{k=1}^{n_c} E_k$ be the total error of the training output and $\widehat{E}_k = E - E_k$ denotes the inverse of E_k. After optimizing the kernels' center positions, we keep these centers unchanged while adjust the width δ_j of hidden units.

$$\delta_j = \delta_j - \eta \nabla \widehat{E}_k; \quad if\ j \in \mathscr{N}_H^k, \quad k \in [1, n_c] \tag{5}$$

The purpose of this adjusting process is to prevent the kernel functions of hidden units belonging to different classes from overlapping each other. As far as the weights are concerned, the final results can be refined through the following adjustment:

$$\mathbf{w}^{L_2} = \mathbf{w}^{L_2} - \eta \nabla E; \quad \mathbf{w}^{L_1} = \mathbf{w}^{L_1} - \eta \nabla^2 E \tag{6}$$

where $w_{kj}^{L_2}$ is the RBF weight connecting the j-th hidden unit and the k-th output unit of the third layer and $w_{ji}^{L_1}$ is the fusion weight connection the i-th image feature and j-th neural of the hidden layer. E is the total error calculated through all training patterns.

3 Multi-scale Features for Region Description

Given a candidate $n_x \times n_y$ image, where each pixel at $(x, y) \in \{1, 2, \cdots, n_x\} \times \{1, 2, \cdots, n_y\}$ has a Γ dimension vector $(l^{(1)}, l^{(2)}, \cdots, l^{(\Gamma)})$ corresponding to the Γ frequencies bands. On each channel $\tau \in \{1, 2, \cdots, \Gamma\}$ the digital image can be represented by a function $l^{(\tau)} = f^{(\tau)}(x, y)$, $l^{(\tau)} \in \{0, 1, \cdots, n_l^{(\tau)} - 1\}$. In the moving window size of $w_1 \times w_1$ from each of the Γ channels separately we have $H_\alpha^{(\tau)} = \frac{|N_\alpha^{(\tau)}|}{w_1^2}$, $\alpha \in \{0, 1, \cdots, n_l^{(\tau)} - 1\}$ where $N_\alpha^{(\tau)}$ is the set of indices to all pixels in region $\mathscr{D}_1$ with $f^{(\tau)}(x, y)$ equals α, and $| \bullet |$ denotes the cardinality of the set. The following first order statistics features are obtained from it $\xi_1 = \sum_{\alpha=0}^{n_l^{(\tau)}-1} \alpha H_\alpha^{(\tau)}$, $\xi_2 = \sum_{\alpha=0}^{n_l^{(\tau)}-1} (\alpha - \xi_1) H_\alpha^{(\tau)}$, $\xi_3 = \sum_{\alpha=0}^{n_l^{(\tau)}-1} [H_\alpha^{(\tau)}]^2$, $\xi_4 = \sum_{\alpha=0}^{n_l^{(\tau)}-1} H_\alpha^{(\tau)} ln\{H_\alpha^{(\tau)}\}$.

Texture is another fundamental feature in defining region and providing information. The Statistical Geometrical Features (SGF) [10] have been reported to perform better than Statistical Grey Level Dependence Matrix (SGLDM) and several other features in texture discrimination [12]. In this paper we extend the SGF to multi-spectral images as a texture descriptor. Firstly we represents the original image $f^{(\tau)}(x, y)$, $(x, y) \in \mathscr{D}_2$ by a set of potentially different binary images as follows.

$$f^{(\tau)}(x, y) = \sum_{\phi=0}^{n_l^{(\tau)}-1} f_b^{(\tau)}(x, y; \phi), \tag{7}$$

$$f_b^{(\tau)}(x, y; \phi) = \begin{cases} 1 & if\ f(x, y) > \phi \\ 0 & otherwise. \end{cases} \tag{8}$$

This transform is bijective and guarantees no loss of information. The next step is to obtain the number of connected regions of 1-valued pixels in the binary

$$\underset{(a)}{\begin{array}{|c|c|c|}\hline 70 & 71 & 36 \\ \hline 71 & 35 & 36 \\ \hline 72 & 35 & 35 \\ \hline \end{array}} \quad \underset{(b)}{\begin{array}{|c|c|c|}\hline 0 & 1 & 1 \\ \hline 1 & 1 & 0 \\ \hline 1 & 1 & 0 \\ \hline \end{array}} \times \underset{(c)}{\begin{array}{|c|c|c|}\hline 1 & 2 & 4 \\ \hline 8 & 16 & 32 \\ \hline 64 & 128 & 256 \\ \hline \end{array}} \quad \underset{(d)}{\begin{array}{|c|c|c|}\hline 0 & 2 & 4 \\ \hline 8 & 16 & 0 \\ \hline 64 & 128 & 0 \\ \hline \end{array}}$$

Fig. 1. A pattern example of LEP descriptor with $\theta_e = 5$

image denoted by $Noc_1^{(\tau)}(\phi)$, and that of 0-valued pixels in the same binary image by $Noc_0^{(\tau)}(\phi)$. Each of the connected regions $\mathcal{C}^{(\tau)}$ has a following measure of irregularity.

$$IRGL(\mathcal{C}^{(\tau)}) = \frac{1 + \sqrt{\pi} max_{i \in \mathcal{C}^{(\tau)}} \sqrt{(x_i - \overline{x})^2 + (y_i - \overline{y})^2}}{\sqrt{|\mathcal{C}^{(\tau)}|}} - 1 \tag{9}$$

where $\overline{x} = \frac{\sum_{i \in \mathcal{C}^{(\tau)}} x_i}{|\mathcal{C}^{(\tau)}|}$, $\overline{y} = \frac{\sum_{i \in \mathcal{C}^{(\tau)}} y_i}{|\mathcal{C}^{(\tau)}|}$. Then the average irregularity of the connected region $\overline{IRGL_0}^{(\tau)}(\phi)$, $\overline{IRGL_1}^{(\tau)}(\phi)$ weighted by set size are obtained and sixteen features are extracted.

In this paper the spatial structure of local region boundary is described using the Local Edge Pattern (LEP) descriptor [8]. A pixel (x_0, y_0) is an edge point if there exist at least one pixel (i, j) belonging to its four neighbors that satisfied $|f^{(\tau)}(x_0, y_0) - f^{(\tau)}(i, j)| > \theta_e$. Fig 1(a) and (b) give a simple pattern that illustrate the pixel labels and the corresponding edge image. Then the binary edge matrix is multiplied by the corresponding binomial weights, as shown in (c) and (d). We sum the resulting values to obtain the LEP value for the center pixel (x_0, y_0). In each window $\mathscr{D}_3$ the normalized LEP histogram can be computed from

$$H_e^{(\tau)}(\gamma) = \frac{|N^{(\tau)}(\gamma)|}{w_3^2}, \quad \gamma \in \{0, 1, \cdots, 511\} \tag{10}$$

where $N^{(\tau)}(\gamma)$ represents the set of pixels with LEP equal γ.

4 Experimental Results and Discussion

Firstly, we test our training method by the Landsat Multi-Spectral Scanner image data coming from UCI. The Landsat satellite data is one of the many sources of information available for a scene.

Table 1. The dataset used in experiment

Dataset	# Classes	# Instances	# Features
Satellite	7	6500	36

The interpretation of a scene by integrating spatial data of diverse types and resolutions including multi-spectral and radar data, maps indicating topography, land use etc. The data set consists of the multi-spectral values of pixels in 3×3 neighborhoods in a satellite image, which contains 36 features (4 spectral bands x 9 pixels in neighborhood) and 6500 records, as shown in table 1. The aim is to predict the class of the central pixel, given the multi-spectral values. We test

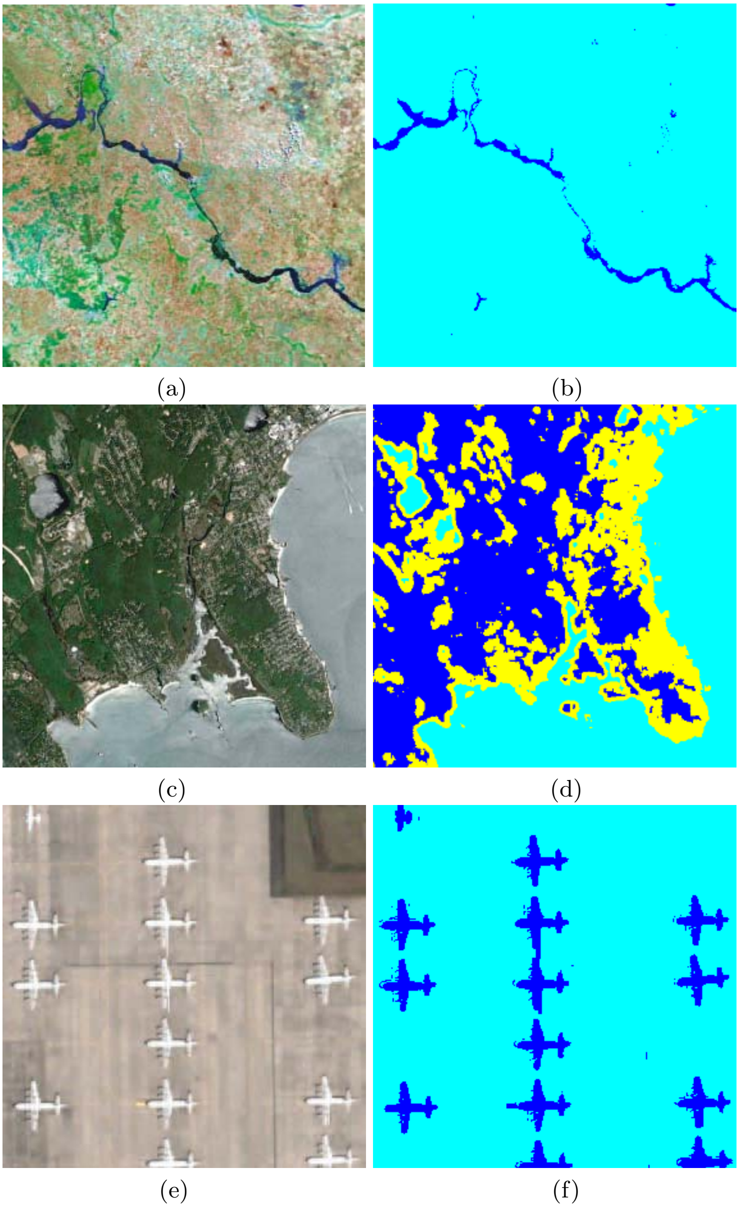


Fig. 2. Original SAR images with different eye altitudes for the experiments (a)$\rho_1 = 356.06mi$. (c)$\rho_2 = 17370ft$. (e)$\rho_3 = 2860ft$. The segmentation results of the corresponding images are (b),(d),(f).

out training method using 5-fold cross-validation. The following table 2 lists the averaged performance on this database obtained by three kinds of training methods. It demonstrates that the proposed approach possesses remarkably stronger classification ability on both the conventional RBF and weighted RBF network.

Table 2. The averaged classification rate on the Landsat Multi-Spectral Scanner image data, where method **A** is the conventional BP training strategy, method **B** choice the hidden center based on clustering and method **C** is the proposed multi-phase training algorithm.

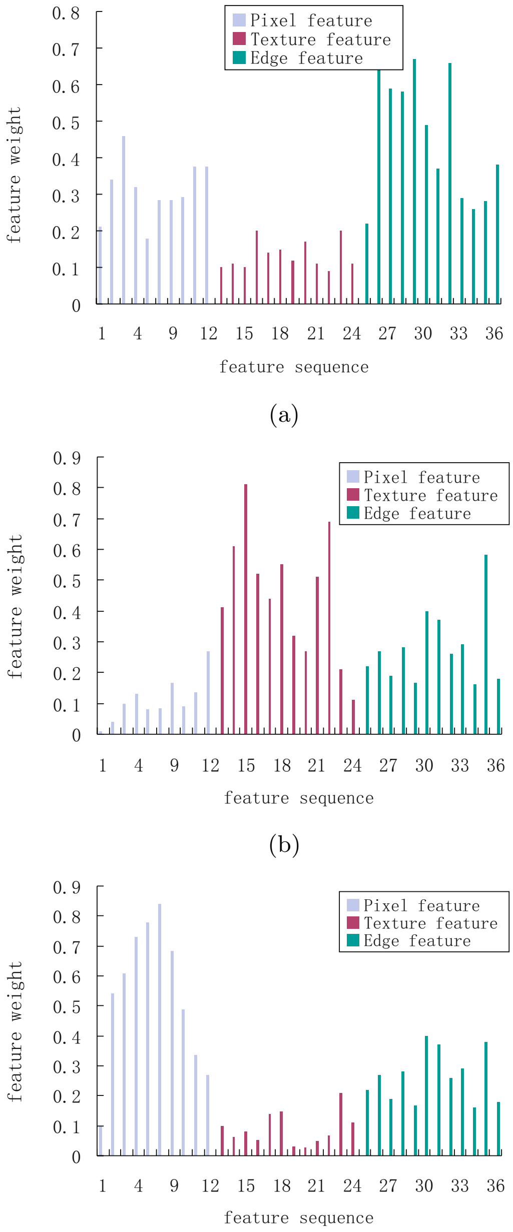
	A	**B**	**C**
Classification Rate using RBF network	87.3 %	90.2 %	**91.3** %
Classification Rate using weighted RBF network	87.1 %	91.3 %	**93.5** %

Next we utilize the proposed method to segment real SAR images. Synthetic Aperture Radar is an active imaging system capable of high-resolution spatial measurement at radar frequencies [7]. Analyzing images generated by SAR systems becomes increasingly important for a variety of applications, such as land-cover mapping, object recognition, forestry and oil-spill detection. In this experiment three 480pi x 480pi JERS-1/SAR images are used in the experiments.

The original images and the corresponding results are presented in Fig. 2. Images in Fig. 2(a),(c),(e) are taken from various altitudes ($\rho_1 > \rho_2 > \rho_3$) with frequency bands $\Gamma = 3$. Sixteen-three binary images (evenly spaced thresholds $\phi = 4, 8, \cdots, 252$ as suggested in [10]) were used in SGF. $w_1 = 5$, $w_2 = 11$, $w_3 = 17$. LEP threshold $\theta_c = 12$. Learning rate $\eta = 0.002$. The kernel width $\delta_j \in [5, 10]$, feature fusion weights and network weights $w_{ji}^{L_1} \in [0,1]$, $w_{kj}^{L_2} \in [0,1]$ are initialized randomly.

After the training process, the weighted RBF network is used to segment the real SAR images. Fig. 2(b),(d),(f) illustrate the obtained segmentation results using pseudo-color. Fig. 2(b) is segmented to water area and land area. The (d) is segmented to residential regions, sea and mountain. In image (f) the contours of planes have been well segmented. The figure 3 shows the final feature weights after the multi-phase training process. The figures show that the proposed network is able to yield proper feature weights according to different situation. The categorical features, including local statistics, texture and edge features, are well evaluated and weighted. The fusion of the multi-scale edge, texture features makes the proposed approach fit for the different situations.

The proposed training strategy contains the characteristics: firstly, the centers can be located as fast as possible because cancelation effect among various parameters is eliminated, which is the major shortcoming of conventional training methods. Secondly, since the moving path of centers are short, it can overcome the probably local minima, hence improves the performance of segmentation.



(a)

(b)

(c)

Fig. 3. After the training process, the distribution of feature weights according to different eye altitudes.(a) feature weights on image ρ_1. (b)feature weights on image ρ_2.(c)feature weights on image ρ_3.

5 Conclusions

This paper proposed a new approach for feature fusion using adaptive RBF neural network, and we applied it to remote sensing images. Multiple features, including radar frequencies distribution, textures, and region boundaries are extracted for the process. A novel multi-phase training strategy is proposed to integrate multiple information sources efficiently. The multi-phase approach with gradient descending is to train the RBF network while distort effects between parameters are eliminated. The neural network yields improved performance which demonstrate the effectiveness when used for different scale SAR images.

Acknowledgements

The research work presented in this paper is supported by National Natural Science Foundation of China, project No.60275010; Science and Technology Commission of Shanghai Municipality, project No. 04JC14014.

References

1. C. Pohl, J. L. Van Genderen: Review article multisensor image fusion in remote sensing: concepts, methods and applications. International Journal of Remote Sensing **19** (5) (1998) 823–854
2. Mpd Jolly, A Gupta: Color and texture fusion: application to aerial image segmentation and GIS updating. Image and Vision Computing **18** (10) (2000) 823–832
3. Carper J.W., Lillesand T.M., Kjefer R.W.: The use of intensity-hue-saturation transformation for merging SPOT panchromatic and multispectral image data. Photogrammetric Engineering and Remote Sensing **56** (4) (1990) 459–467
4. Jia Yonghong: Fusion of Landsat TM and SAR image based on principal component analysis. Remote Sensing Technology and Application **13** (3) (1998) 46–49.
5. Bruno Aiazzi, Luciano Alparone, and Andrea Garzelli: Context-driven fusion of high spatial and spectral resolution images based on oversampled multiresolution analysis. Transactions on Geoscience and Remote sensing **40** (2002) 2300–2312.
6. Michie, D.Spiegelhalter, D.Taylor: Machine Learning, Neural and Statistical Classification, New York, Ellis Horwood (1994)
7. C.J.Oliver, S.Quegan: Understanding SAR Images. Boston, Artech House (1998)
8. K.M.Chen, S.Y.Chen: Color Texture segmentation using feature distributions. Pattern Recognition Letters **23** (2002) 755–771
9. K.Z. Mao: RBF Neural Network Center Selection Based on Fisher Ratio Class Separability Measure. IEEE trans, Neural Networks **13** (2002) 1211–1217
10. Chen Y.Q., Nixon M.S., Thomas D.W.: Statistical Geometrical Features for Texture Classification. Pattern Recognition **28** (4) (1995) 537–552
11. James J.S. and T.J.Mcintire: A recurrent neural network classifier for improved retrievals of areal extent of snow cover. IEEE trans on Geosciene and remote sensing. **39** (10) (2001) 2135–2147
12. Singh M., Singh S.: Spatial texture analysis: a comparative study. Proc. 15th International Conf. on Pattern Recognition (ICPR'02) **1** (2002) 676–679

13. Sami M.A., W.L.Jones, J.D. Park and S.M.Ferguson: A neural network algorithm for sea ice edge classification. IEEE trans on Geosciene and remote sensing. **35** (4) (1997) 817–826
14. F.Schwenker, H.A.Kestler, G.Palm: Three learning phases for radial-basis-function networks. Neural Networks **14** (2001) 439-458
15. Francesco L., Marco S.: Effcient training of RBF neural networks for pattern recognition. IEEE trans on neural networks **12** (5) (2001) 1235–1241
16. Dietrich Wettschereck, Thomas Dietterich: Improving the performance of Radial Basis Function networks by learning center locations. Advance in Neural Information Processing System 4 Morgan Kaufmann Publisher (1992)
17. Chitra P., Marimuthu P., Daniel Ralph, Chris Manzie: Effects of moving the centers in an RBF network. IEEE trans on Neural Networks **13** (6) (2002) 1299–1307

Active Contour with Neural Networks-Based Information Fusion Kernel

Xiongcai Cai[1,3] and Arcot Sowmya[1,2,3]

[1] School of Computer Science and Engineering,
The University of New South Wales, Sydney, NSW 2052, Australia
[2] Division of Engineering, Science and Technology, UNSW Asia, Singapore
[3] National ICT Australia, Locked Bag 6016, NSW 1466, Australia
{xcai, sowmya}@cse.unsw.edu.au

Abstract. This paper proposes a novel active contour model for image object recognition using neural networks as a dynamic information fusion kernel. It first learns feature fusion strategies from training data by searching for an optimal fusion model at each marching step of the active contour model. A recurrent neural network is then employed to learn the fusion strategy knowledge. The learned knowledge is then applied to guide another linear neural network to fuse the features, which determine the marching procedures of an active contour model for object recognition. We test our model on both artificial and real image data sets and compare the results to those of a standard active model, with promising outcomes.

1 Introduction and Related Work

Automatic object extraction and recognition from images is a fundamental problems in computer vision. The general approaches adopted include thresholding techniques, edge-based methods, region-based techniques, and connectivity-preserving relaxation methods. Traditional image segmentation approaches are driven by the intrinsic contrast between objects and their background, captured by individual low level features such as intensity, color, gradient or textures. However, these approaches fail when there is no distinction in individual features between objects and their background. Even when multiple features are fused, a static fusion model can still significantly degrade the effect of the fused features. Learning of dynamic information fusion knowledge has become an important topic in the area of object extraction and recognition within the field of computer vision.

Active contours are used to detect objects in a given image u_0 using techniques of curve evolution. The basic idea is to deform the curve to the boundary of the object starting with an initial curve C, under some constraints from the image u_0. To address curve evolution, deformable contour models or snakes were first presented [1] for detection and localisation of boundaries. Cohen [2] uses the balloon model to reduce the requirement of initialisation of the snake model. This has been improved [3] using a geodesic formulation in a Riemannian space for

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 324–333, 2006.

active contours derived from the image content. Cohen and Kimmel [4] describe a shape modeling method by interpretation of the snake as a path of minimal cost which is solved using numerical methods. Level set method has been utilised for shape modelling [5] because it allows for detection of automatic topology changes, cusps and corners. Geman and Jedynak [6] present an active testing model to reduce uncertainty in tracking roads in satellite images using entropy and statistical inference. The approaches only work for low level segmentation and are not suitable for higher level object extraction or recognition due to their inability to learn and utilise prior object knowledge.

We have recently developed a method to introduce control parameters into the speed function of level set methods and utilised a genetic algorithm to tune those parameters to adjust the effect of intensity and gradient features and force the marching of the active contours to stop at the of object boundaries [7][8].

Chan *et al.* extend the scalar Chan-Vese algorithm for active contours [9] to the vector value case. The model minimises a Mumford-Shah functional over the length of the contour, as well as the sum of the fitting error over each component of the vector-valued image. Multiple features were simply combined with manually selected and fixed weights and there was no usage of prior object knowledge, therefore not suitable for recognition. Trainable fusion strategies are essential for the success of a robust object recognition system.

In this paper, we propose a novel active contour model for image object extraction using a neural network fusion kernel, which has the ability to choose and weight features to self-adapt its marching procedures by using feature fusion strategy knowledge learned from training data. It first searches for an optimal fusion model or weight value vector for each marching step. It then utilises the search result to learn feature fusion strategies from training data. Then the learned knowledge is used to guide the marching procedures for object extraction. Our major contribution is the embedding of information fusion learning using neural networks in the active contour model for object extraction and recognition.

The paper is organized as follows. In section 2, we introduce the active contour model. An information fusion kernel using neural networks is described in section 3. Experiments are described and analysed in section 4. Conclusions are presented in section 5.

2 Active Contour Model

Let Ω be a bounded open subset of $\Re^2$, with $\partial\Omega$ the boundary. Let u_0 be a given image such that $u_0 : \Omega \rightarrow \Re$. Let $C(s) : [0,1] \rightarrow \Re^2$ be a parameterised C^1 curve. In [5], the classical level set boundary is defined as the zero level set of an implicit function $z = \phi(x, y, t)$ defined on the entire image domain. The contour at time t must satisfy the function $\phi(x, y, t) = 0$.

There are different level set formulations [5][3][10]. We follow the vector-valued version of the C-V model [10]. Let $u_{0,1}$ be the i_{th} channel of an image on Ω, with i = 1, $\cdots$, N channels, and C the evolving curve. Each channel would present different characteristics of the same image. Let $c^+ = (c_1^+, \cdots, c_N^+)$ and $c^- = (c_1^-, \cdots, c_N^-)$ be two unknown constant vectors. The energy function of the active contour model is defined as follows:

$$F(\bar{c^+}, \bar{c^-}, \phi) = \mu.Length(C) + \int_{inside(C)} \frac{1}{N} \sum_{i-1}^{N} \lambda_i^+ |\mu_{0,i}(x,y) - c_i^+|^2 dxdy$$
$$+ \int_{outside(C)} \frac{1}{N} \sum_{i-1}^{N} \lambda_i^- |\mu_{0,i}(x,y) - c_i^-|^2 dxdy \quad (1)$$

Fitting the above energy function into a level set framework to minimise F with respect to ϕ, we get the following Euler-Lagrange equation for ϕ:

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon [\mu.div(\frac{\bigtriangledown \phi}{|\bigtriangledown \phi|}) - \frac{1}{N} \sum_{i-1}^{N} \lambda_i^+ |\mu_{0,i} - c_i^+|^2 + \frac{1}{N} \sum_{i-1}^{N} \lambda_i^- |\mu_{0,i} - c_i^-|^2] \quad (2)$$

In the approach using the above equation [10], manually tuned and fixed weight-parameters λ_i are employed to combine channels of the same image.

3 Information Fusion Kernel

3.1 The Proposed Model

Let m_i^+ be the mean of *i*th feature values of objects and m_i^- be those of the background learned from training data. We utilise the mean of feature values inside and outside the object boundary and embed those statistics *a priori* into the level set equation. Then based on equation 2, ϕ^{n+1} can be defined as follows:

$$\phi^{n+1} = \phi^n + \triangle t.\delta_\epsilon [\mu.div(\frac{\bigtriangledown \phi}{|\bigtriangledown \phi|}) - \frac{N - \sum_{i=1}^{N} e^{-|c_i^+ - m_i^+|}}{N} \sum_{i=1}^{N} \lambda_i^+ |\mu_{0,i} - c_i^+|^2$$
$$+ \frac{N - \sum_{i=1}^{N} e^{-|c_i^- - m_i^-|}}{N} \sum_{i=1}^{N} \lambda_i^- |\mu_{0,i} - c_i^-|^2] \quad (3)$$

The model searches for the best vector-valued approximation taking only two values, the constant vectors $\hat{c^+}$ and $\hat{c^-}$. The active contour is the boundary between these two regions. The energy balances the lengths of the contours in the images, shown as the first terms in the square bracket in Equation 3. It fits u_0 to $\hat{c^+}$ and $\hat{c^-}$ in the data driven terms shown as the last two terms in the square bracket in Equation 3. It also fits $\hat{c^+}$ and $\hat{c^-}$ to m_i^+ and m_i^-, averaged over all features.The *prior* fitting terms:

$$N - \sum_{i=1}^{N} e^{-|c_i^+ - m_i^+|}, N - \sum_{i=1}^{N} e^{-|c_i^- - m_i^-|} \quad (4)$$

are responsible for reducing the energy when the marching contour proceeds to the true object boundary, in order to fit the extracted object into the *prior* statistics learned from the training data. The weight parameters λ_i are essential to the success of the model. They adjust the individual features using information fusion, which is obtained by a learner. In this case, it is a recurrent neural network, trained on training data. Once the parameter model is learned, it is employed to dynamically construct the linear information model.

3.2 The Fusion Kernel

To learn the linear information fusion model, our algorithm consists of two phases following the automatic parameter tuning method in [7]. It first searches the parameter space for an optimal parameter value vector with respect to the performance of the level set model with the help of references in the training data. Secondly, the optimal parameter value vector obtained as well as the feature vectors are used to discover the relationship between them. An overview of the information fusion algorithm is depicted in Fig. 1.

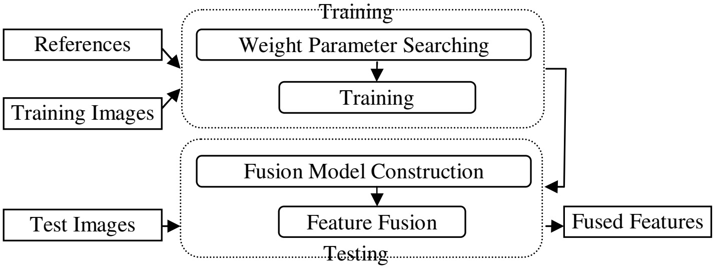


Fig. 1. Automatic Information Fusion Overview

Parameter Search. The goal of parameter search is to find a parameter value vector, with which each marching step of the level set method can achieve optimal performance. Let λ be the parameter to search and ϕ^{n+1} the level set after marching using λ; we seek a $\hat{\lambda}$ such that the Euclidean distance between ϕ^{n+1} and ϕ^{ref} is minimum. Thus, $\lambda_{i,n}$ is obtained by an optimisation procedure with respect to the following conditions:

$$\lambda_{i,n} = \arg\min_{\lambda} \sum_{x=1,y=1}^{N,M} |\phi_{x,y}^{n+1} - \phi_{x,y}^{ref}| \tag{5}$$

where $\phi_{x,y}^{ref}$ is the level set value of the signed distance surface for the reference image at location (x,y).

Parameter Learning. The learning problem can be formulated as follows: let **x** be the mean feature value vector for the inside and outside of the object and **y**

the weight-parameter value vector. Our goal is to learn the relationship between **x** and **y** for each marching step during the level set evolution.

Neural networks based learning methods provide a robust approach to approximating vector-valued target functions. For certain types of problems, such as learning to interpret complex sensor data, neural networks are among the most effective learning methods currently known [11]. Moreover, since the parameters are dynamically selected for each marching step, the marching steps are dependent on the previous steps and the temporal patterns across marching steps must be modeled. We assume that the current marching step is dependent on only the immediately previous step and only first order temporal constraints are considered. The Elman recurrent neural network [12] provides such a solution for parameter learning. The network architecture is as described in Fig. 2.

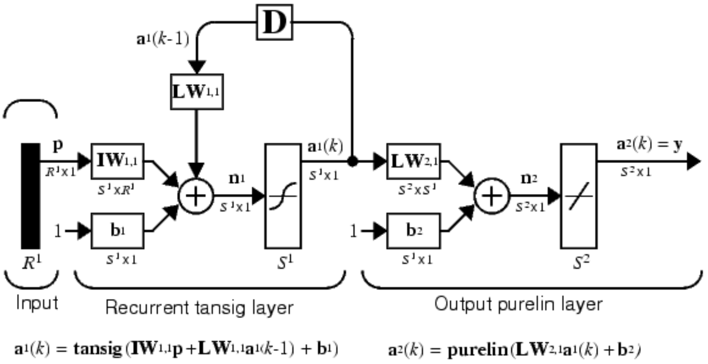


Fig. 2. Recurrent Neural Network. In layer 1, a R^1 length input vector P is connected to a neuron input through the weight matrix IW. The neuron has a summer that gathers its weighted inputs, bias b_1 and the output of previous time step $a^1(k-1)$ to form its own scalar output n^1. Then the first layer outputs form a column vector $a^1(k)$ by a transfer function S^1, which is then input to layer 2 to be weighted by LW and added bias b^2, and then to form the outputs of the network $a^2(k)$ by another transfer function S^2. "tansig" and "purelin" are the transfer function for S^1 and S^2 respectively, shown in the bottom of the figure.

This Elman recurrent connection allows both to detect and to generate time-varying patterns. The network used is a two-layer network with feedback from the first-layer output to the first layer input. It has tangent sigmoid neurons in its hidden layer, and linear neurons in its output layer. This combination is special in that two-layer networks with these transfer functions can approximate any function (with a finite number of discontinuities) with arbitrary accuracy. The only requirement is that the hidden layer must have a sufficient number of neurons. The Elman network differs from conventional two-layer networks in that the first layer has a recurrent connection. The delay in this connection stores values from the previous time step, which can be used in the current time step. Thus, even if two Elman networks, with the same weights and biases, are given identical inputs at a given time step, their outputs can be different due to different feedback states. Because the network can store information for future reference, it is able to learn temporal patterns as well as spatial patterns.

3.3 Linear Information Fusion

Given weight-parameter values, fusion is modeled as a single layer linear neural network, where weight values are obtained from the recurrent neural network described in section 3.2. The linear neural network for information fusion is shown in Fig.3. The inputs to this neural network are the mean feature values inside and outside of the object. The output is the data driven energy term fused over all features.

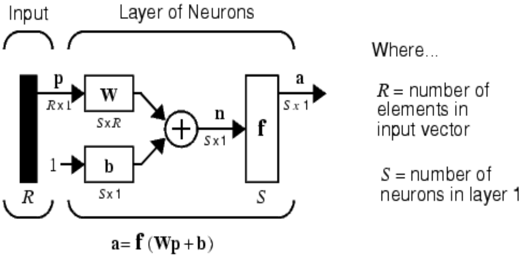

Input
Layer of Neurons
Where...
p
Rx1
W
SxR
n
Sx1
f
a
Sx1
1
b
Sx1
R
S
R = number of elements in input vector
S = number of neurons in layer 1
a= f (Wp+b)

Fig. 3. Linear Neural Network. P is an R length input vector, W is an SxR matrix, **a** and **b** are S length vectors. The neuron layer includes the weight matrix, the multiplication operations, the bias vector b, the summer, and the transfer function boxes.

4 Experiments

4.1 Experimental Setup

Optimization techniques are used to find a set of design parameters, that can in some way be defined as optimal. In our case, it is the maximisation of the level set model that is dependent on the parameters λ. To assure stability of the algorithm, the optimisation is subject to constraints in the form of inequality constraints: $0 \leq \lambda_i \leq 1$. We use Sequential Quadratic Programming optimisation method implemented in the Optimization Toolkit of Matlab in our experiments. The output of the optimization is a tuple of image feature values and optimised parameter values, which are also the input to the recurrent neural network. We use the neural networks implementation from the Neural Network Toolkit of Matlab. In our experiments, we set the parameters for learning neural networks as shown in Table 1. For the level set model, we choose a time step $\triangle t = 0.1$.

The experiments were performed on both synthetic and real images with different types of contours, shapes and textures. The active contours evolving in

Table 1. Parameter Setting

Parameter	Value(s)

Transfer function	TF1 = Hyperbolic tangent sigmoid
	TF2 =Linear

Backpropagation network training function	BTF = Levenberg-Marquardt
Backpropagation weight learning function	BLF = Gradient descent with momentum
Performance function	PF = Mean squared error
Learning rate	Learning rate:lr = 0.05 Learning rate increment = 1.05

the original image are the associated piecewise-constant approximation. While three types of features, namely local standard deviation, local entropy and gradient, were fused in our experiments, the proposed approach can be used for more features. We ran the level set method with 200 iteration steps for each image. Notice that the optimisation ouput is based on marching steps and they are used as training instances and test instances. This means that there are 200 training instances or test instances from each image, which reduces the number of necessary training and test images significantly. This is important because gathering enough references for training data can be very expensive in most application areas, such as medical image and remotely sensed image processing.

For further evaluation of the proposed algorithm, two standard active contour algorithms, Active Contour without Edges [9] and Geodesic Active Contours [3], were carried out on the test data. The result of theproposed algorithm was then compared to those of the standard active contour algorithms to demonstrate the improvements of the proposed method.

4.2 Experimental Results

In Fig. 4, we show how our model works on synthetic images, where the objects are automatically detected according to the training objects. There are two objects in the images. One of the objects has relative high intensity value but very low feature values, while the other has almost the same intensity value as the background but high feature values. Notice that when extracting both objects, the objects in this image cannot be represented by any single type of feature and thus cannot be detected without information fusion as described below in Fig. 5. Furthermore, since the shapes in image data are different from each other, they cannot be extracted properly by using a single prior shape model either.

In Fig. 5, we present results of standard active contours performed on Image 1 of Fig. 4. Geodesic Active Contours was executed on the original test image. It could merely extract the object on the left upper of the test image due to its higher boundary gradient value, but missed the other object. For the algorithm of Active Contour without Edges, we ran it on the intensity image and three feature images of the test image. For the intensity image and entropy feature image, we got similar result as that of Geodesic Active Contours. For the standard deviation

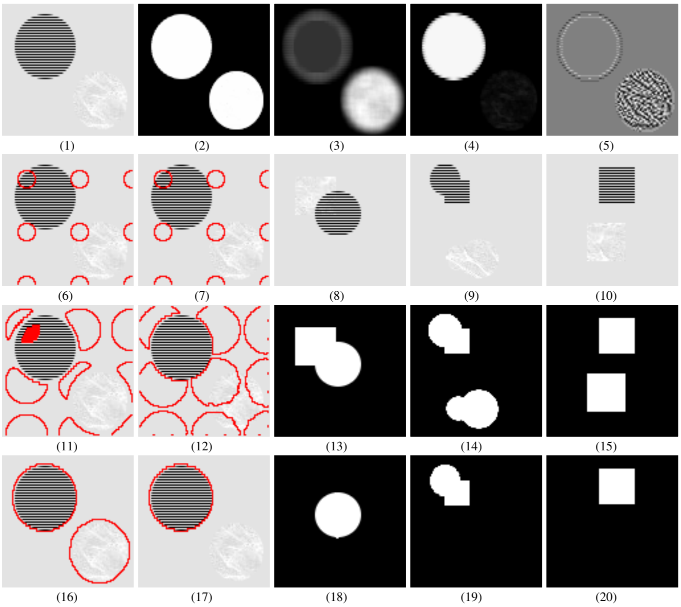


Fig. 4. Images 8-10 show the training images, Images 13-15 their reference for extracting 2 objects and Images 18-20 their reference for extracting one object. Image 1 is a test image, Image 2 its reference (for comparision) and Image 3-5 show feature images for Image 1. Images 6, 11, 16 show the initial, middle and final steps of extracting 2 objects in Image 1, after learning on Images 13-15. Images 7, 12, 17 show the initial, middle and final steps of extracting 1 object in Image 1, after training on Images 18-20.

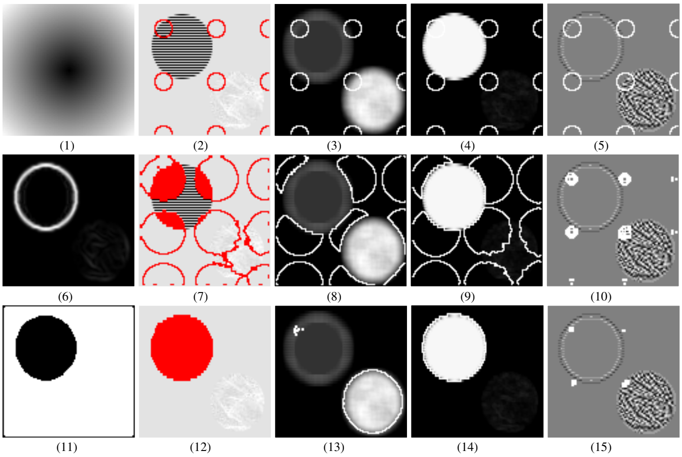
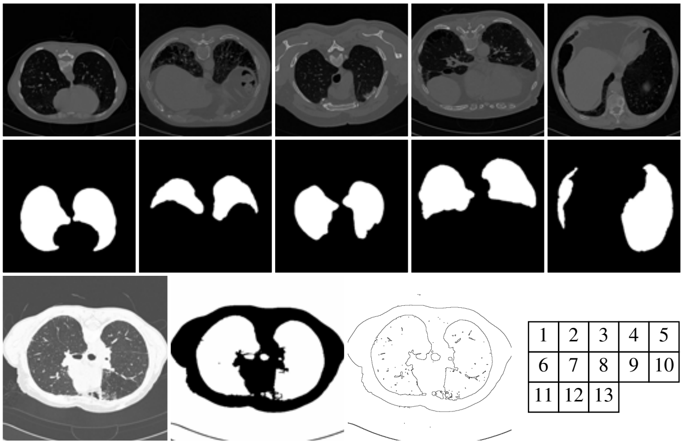


Fig. 5. Experimental results of standard active contours on Image 1 of Fig. 4. Images 1, 6, 11 show the initial, edge and final result image of Geodesic Active Contours. Images 2-5, 7-10 and 12-15 present the results of Active Contour without Edges. Images 2-5 show the intensity, standard deviation, entropy and gradient images. Images 7-10 show the middle step images and Image 12-15 the final result images, respectively.



1	2	3	4	5
6	7	8	9	10
11	12	13		

Fig. 6. Experimental results on real medical image data from LMIK dataset [13]. Images 1-5 are the original training images whose references are shown as Images 6-10 respectively. Image 11 and 13 show a test and result image, with the reference image shown in Image 12 for comparision.

feature image, it only extracted the object in the right lower contour and missed the other. The worst case happened on the gradient feature image, where Active Contour without Edges hardly extracted any object.

These experimental results show that neither of the standard active contours has the ability to extraction both objects in the test image at the same time. However, as shown in the Figure 4, the proposed algorithm is able to extract and recognise either both objects or a single object by using dynamic feature fusion, according to the training data used.

In Fig. 6, we show our model trained to recognise lung boundares in real medical images. The training data consists of 5 CT images containing lung slices. The proposed model is trained with the parameters shown in Table 1. The test images are different from the training data.

5 Conclusion

This paper proposes an approach to learn a feature fusion model for the active contour algorithm based on level set method and neural networks. The model is not based on heuristically applying naive features to achieve extraction, but rather a learnable information fusion kernel to combine features. Furthermore, including automatic initialisation, the model is fully automatic without the use of heuristic parameters for feature combination and other manual interaction. The numerical results have demonstrated the feasibility of the proposed method.

Acknowledgement

National ICT Australia is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council.

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision (1988) 321–331
2. Cohen, L.: On active contour models and balloons. CVGIP Image Understanding **53** (1991)
3. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: ICCV'95, Cambridge, USA (1995) 694–699
4. Cohen, L.D., Kimmel, R.: Global minimum for active contour models: A minimal path approach. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (1996) 666–673
5. Malladi, R., Sethian, J.A., Vemuri, B.C.: Shape modeling with front propagation: a level set approach. IEEE Transactions on Pattern Analysis and Machine Intelligence **17**(2) (1995)
6. Geman, D., Jedynak, B.: An active testing model for tracking roads in satellite images. IEEE Trans. Pattern Anal. Machine Intell. **18**(1) (1996)
7. Cai, X., Sowmya, A., Trinder, J.: Learning parameter tuning for object extraction. ACCV2006, Lecture Notes in Computer Science **3851** (2006) 868–877
8. Cai, X., Sowmya, A., Trinder, J.: Learning to recognise roads from high resolution remotely sensed images. In: The 2nd International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Melbourne, IEEE (2005) 307–312
9. Chan, T., Vese, L.: Active contours without edges. IEEE Transactions on Image Processin **10**(2) (2001) 266–277
10. Chan, T., Sandberg, B., Vese, L.: Active contours without edges for vector valued images. Journal of Visual Communication and Image Representation **11** (2000) 130–141
11. Mitchell, T.: Machine Learning. McGraw-Hill (1997)
12. Elman, J.: Finding structure in time. Congitive Science **14** (1990) 179–210
13. Rudrapatna, M., Sowmya, A., Zrimec, T., Wilson, P., Kossoff, G., Lucas, P., Wong, J., Misra, A., Busayarat, S.: Lmik learning medical image knowledge: An internet-based medical image knowledge acquisition framework. In: Internet Imaging. Volume 5305., San Jose, CA, USA (2004) 307–318

A Novel Split-and-Merge Technique for Error-Bounded Polygonal Approximation

Bin Wang[1] and Chaojian Shi[1,2]

[1] Department of Computer Science and Engineering, Fudan University,
Shanghai, 200433, P.R. China
[2] Merchant Marine College, Shanghai Maritime University,
Shanghai, 200135, P.R. China
wangbin.cs@fudan.edu.cn, cjshi@shmtu.edu.cn

Abstract. How to use a polygon with the fewest possible sides to approximate a shape boundary is an important issue in pattern recognition and image processing. A novel split-and-merge technique(SMT) is proposed. SMT starts with an initial shape boundary segmentation, split and merge are then alternately done against the shape boundary. The procedure is halted when the pre-specified iteration number is achieved. For increasing stability of SMT and improving its robustness to the initial segmentation, a ranking-selection scheme is utilized to choose the splitting and merging points. The experimental results show its superiority.

1 Introduction

Error-bounded polygonal approximation can be stated as follows: given a shape boundary, approximate it by a polygon with the minimal number of line segments such that the approximation error is no more than a pre-specified tolerance. The goal of error-bounded polygonal approximation is to capture the essence of the shape boundary with the fewest possible polygonal segments [1]. Error-bounded polygonal approximation not only provide a compact shape representation, but also facilitate feature extraction for further image analysis. Therefore it plays an important role in pattern recognition and shape data compression.

In recent decades, many methods have been proposed for error-bounded polygonal approximation. They can be classified into two categories: local-search-based methods and global-search-based methods. Sequential tracing, split and merge techniques are widely used local-search-based methods. For sequential methods, Sklansky and Gonzales [2] proposed a scan-along scheme to start from a point for finding the longest line segments sequentially. Ray and Ray [3] proposed a method of determining the longest possible line segments with the minimum possible error. Sequential methods are simple and fast, however the quality of their final solutions depends on the choice of the starting point. Ramer [4] proposed a split-based method. It is a recursive procedure which starts from an initial boundary segmentation. The iterative procedure repeatedly splits the shape boundary at the point with the farthest distance from the corresponding segment until the approximation error is smaller than the error tolerance. The

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 334–342, 2006.

disadvantage of split method is that the approximation result is sensitive to the initial boundary segmentation.

The common idea in sequential tracing and split method is to choose boundary point to be vertices of the polygonal approximation. While, different from them, Pikaz and Dinstein [5] proposed a merge method which did polygonal approximation in opposite direction. Its main idea is that: initially consider all the boundary points as vertexes and a procedure is repeated to remove the boundary point which will cause minimal increase in the approximation error until the desired approximation error is reached. Like the sequential tracing method, it also exists the problem of depending on the starting point.

Some global-search-based methods such as genetic algorithms (GA) have also been proposed for error-bounded polygonal approximation. GA is based on stochastic search which simulates the biological model of evolution [7]. Yin [6] proposed a GA-based method for polygonal approximation. In the proposed method, a chromosome is used to represent a polygon by a binary string. Each bit, called a gene, denotes a point on the shape boundary. Three genetic operators, including selection, crossover and mutation, are designed to obtain promising polygonal approximations. Although the initial population is randomly generated, the final solution does not depend on the initial solutions because of the population-search and evolution scheme. Therefore, compared with the former mentioned methods, GA-based method is statable. However, the population-search scheme will require higher cost of space and time. Therefore, they are not fit for practice application.

In this paper, a split-and-merge technique (SMT) is proposed for error-bounded polygonal approximation. SMT starts from an initial shape boundary segmentation, split and merge are then alternately done against the shape boundary. The procedure is halted when the pre-specified iteration number is achieved. For overcome the problem of the traditional methods' dependence on the initial segmentation, during the split and merge process, a ranking-selection scheme is utilized for the choice of the boundary points. Three benchmark shape boundaries are used to test the effectiveness of SMT and experimental results show that it outperforms the traditional split method and GA-based method.

2 Problem Definition

A 2D shape boundary is represented as an ordered sequence of points $C = \{p_1, p_2, \ldots, p_N\} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, where $p_{i+N} = p_i$ and N is the number of the points on the shape boundary. Let $\widehat{p_ip_j} = \{p_i, p_{i+1}, \ldots, p_j\}$ denotes the arc which starting from the point p_i and continuing to point p_j in the clockwise direction along the shape boundary. Let $\overline{p_ip_j}$ denote the line segment connecting points p_i and p_j. The approximation error between $\widehat{p_ip_j}$ and $\overline{p_ip_j}$ is defined as follows:

$$e(\widehat{p_ip_j}, \overline{p_ip_j}) = \sum_{p_k \in \widehat{p_ip_j}} (y_k - a_{ij}x_k - b_{ij})^2/(1 + a_{ij}^2) \tag{1}$$

where $a_{ij} = (y_j - y_i)/(x_j - x_i)$ and $b_{ij} = y_i - a_{ij}x_i$. The polygon V approximating the shape boundary $C = \{p_1, p_2, \ldots, p_N\}$ is an ordered set of line segments $V = \{\overline{p_{t_1}p_{t_2}}, \overline{p_{t_2}p_{t_3}}, \ldots, \overline{p_{t_{M-1}}p_{t_M}}, \overline{p_{t_M}p_{t_{M+1}}}\}$, such that $t_1 < t_2 < \ldots < t_M$ where $t_i \in \{1, 2, \ldots, N\}$. The approximation error between shape boundary C and polygon V is defined as

$$E(V, C) = \sum_{i=1}^{M} e(\widehat{p_{t_i}p_{t_{i+1}}}, \overline{p_{t_i}p_{t_{i+1}}}). \tag{2}$$

Then the error-bounded polygonal approximation is defined as follows: given a shape boundary $C = \{p_1, p_2, \ldots, p_N\}$ and a pre-specified tolerance error ε. Let S be the set of all the polygons which approximate the shape boundary C. Let $SP = \{V \mid V \in S \wedge E(V, C) \leq \varepsilon\}$, Find a polygon $P \in SP$ such that

$$\mid P \mid = \min_{V \in SP} \mid V \mid, \tag{3}$$

where $\mid \cdot \mid$ denotes the cardinality of the set.

3 Brief Review of Split and Merge Techniques

Split technique is a recursive procedure which starts from an initial curve segmentation which divided the shape boundary into two sections. At each iteration, a split operator is conducted to divide the segment into two sections at the selected boundary point. The iteration process is repeated until the approximation error is smaller than the tolerance error. Assume that the shape boundary C has been segmented into M arcs $\widehat{p_{t_1}p_{t_2}}, \ldots, \widehat{p_{t_{M-1}}p_{t_M}}, \widehat{p_{t_M}p_{t_1}}$ through $k-1$ iterations, where p_{t_i} is the division point. Then at k-th iteration, the split operation is as follows: for each point $p_i \in \widehat{p_{t_j}p_{t_{j+1}}}$, $j = 1, 2, \ldots, M$, calculate the distance between it to the corresponding chord $D(p_i) = d(p_i, \overline{p_{t_j}p_{t_{j+1}}})$, where $d(p_i, \overline{p_{t_j}p_{t_{j+1}}})$ is the perpendicular distance from point p_i to the line segment $\overline{p_{t_j}p_{t_{j+1}}}$. Find a point p_u on the shape boundary which satisfies $D(p_u) = \max_{p_i \in C} D(p_i)$. Suppose that $p_u \in \widehat{p_{t_k}p_{t_{k+1}}}$. Then the arc $\widehat{p_{t_k}p_{t_{k+1}}}$ is segmented at the point p_u into two arcs $\widehat{p_{t_k}p_u}$ and $\widehat{p_u p_{t_{k+1}}}$. Through split operation, the boundary point p_u is selected as the polygon's new vertex. Fig. 1 gives an example to show a split operation.

Different from split method, merge technique produces polygonal approximation in opposite direction. It starts from an initial polygon which considers all the boundary points as vertexes. At each iteration, a merge operation is done to combine the selected two adjacent arcs into a single one. While the approximation error does not exceed the tolerance error, the procedure is repeated. The detail of merge operation is as follows: suppose that the boundary C has been segmented into M arcs $\widehat{p_{t_1}p_{t_2}}, \ldots, \widehat{p_{t_{M-1}}p_{t_M}}, \widehat{p_{t_M}p_{t_1}}$, where p_{t_i} is a division point. Then a merge operation against the boundary is defined as: for each division point p_{t_i}, calculate the distance to the line segment which connect its two adjacent points $Q(p_{t_i}) = d(p_{t_i}, \overline{p_{t_{i-1}}p_{t_{i+1}}})$. Select a segment point p_{t_j} which

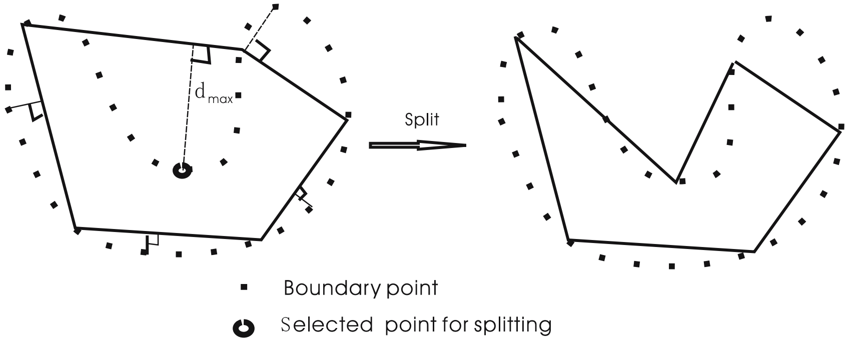


Fig. 1. Split operation

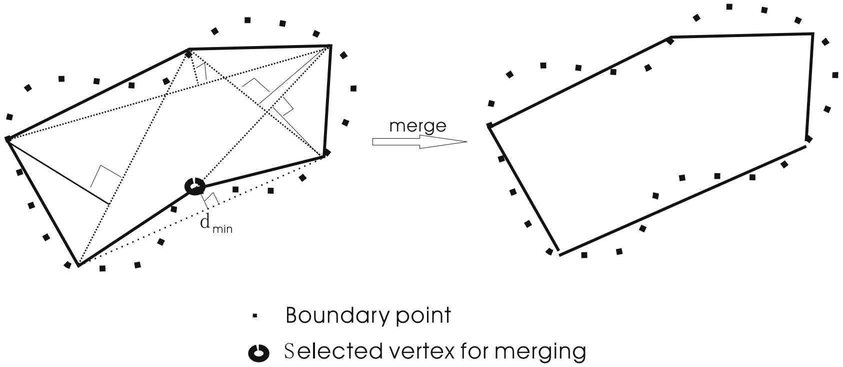


Fig. 2. Merge operation

satisfies $Q(p_{t_j}) = \min_{p_{t_i} \in V} Q(p_{t_i})$, where V is the set of the current division points. Then two arcs $\widehat{p_{t_{j-1}}p_{t_j}}$ and $\widehat{p_{t_j}p_{t_{j+1}}}$ are merged into a single arc $\widehat{p_{t_{j-1}}p_{t_{j+1}}}$. The division point p_{t_j} is removed from the set of the current division points. Fig. 2 gives an example to show a merge operation.

4 Ranking-Selection Scheme

Ranking-selection scheme is initially proposed by Baker [8]. Its purpose is to solve the problem of GA's premature convergence. Selection is an important operator of GA. The traditional GA adopts fitness-proportion-selection scheme. Its main idea is as follows: for a population of M individuals, assume that $f_1, f_2, \ldots, f_M$ are their fitness values, the i-th individual will be assigned a selection probability $\rho_i = f_i / \sum_{i=1}^{M} f_i$. The disadvantage of this selection scheme is that it may lead to premature convergence. For overcome this problem, rank-selection scheme does not determined the selection probability on the fitness values directly. It firstly sorts all the individuals to form an ordered sequence by their finess values, the best individual is in the position one and the worst one is in the position M. The selection probability of the individuals is the function of the position. Let $P = \{x_1, x_2, \ldots, x_M\}$ represent a sorted population, i.e. we have $f(x_1) \geq f(x_2) \geq \ldots \geq f(x_M)$, where $f(x_i)$ is the fitness function of the individual x_i. Then the selection probability $\rho(x_i) = g(i)$, while $g(i)$ is the function of position i and it must satisfies the following constraint conditions: (1) $g(1) \geq g(2) \ldots \geq g(M)$ and (2) $\sum\limits_{i=1}^{M} g(i) = 1$. The function can be linear or non linear.

5 Proposed Method

5.1 Split Operation with Ranking-Selection

In section 3, traditional split operation is introduced, it always select the boundary point with the farthest distance for splitting. This selection scheme will lead to only obtaining local optimal solution. Here we propose a novel split operation which use the ranking-selection scheme. Assume that the shape boundary $C = \{p_1, p_2, \ldots, p_N\}$ has been segmented into M arcs $\widehat{p_{t_1}p_{t_2}}, \ldots, \widehat{p_{t_{M-1}}p_{t_M}}, \widehat{p_{t_M}p_{t_1}}$. Let $W = C - \{p_{t_1}, p_{t_2}, \ldots, p_{t_M}\} = \{p_{v_1}, p_{v_2}, \ldots, p_{v_{N-M}}\}$. For each point $p_{v_i} \in W$, calculate the distance between it to its corresponding chord $d(p_{v_i})$. The set W is then sorted into an ordered set $\{p_{u_1}, p_{u_2}, \ldots, p_{u_{N-M}}\}$ such that $d(p_{u_1}) \geq d(p_{u_2}) \geq \ldots \geq d(p_{u_{N-M}})$. The selection probability $\rho(p_{u_i})$ for point p_{u_i} is defined as

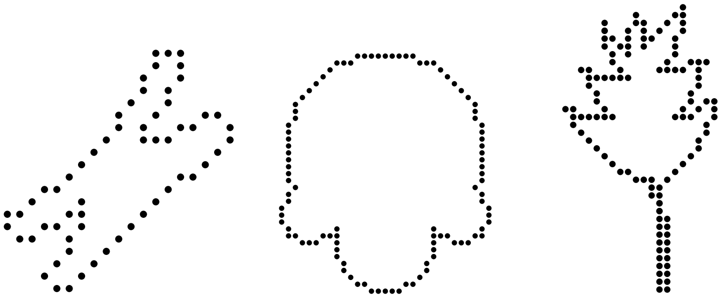
$$\rho(p_{u_i}) = \frac{i^{-r}}{\sum\limits_{j=1}^{N-M} j^{-r}}, \tag{4}$$

where r is the parameter used to adjust the probability distribution. According to the selection probability, choose a point p_{u_i} from the set W and divide the corresponding arc into two sections.

5.2 Merge Operation with Ranking-Selection

The traditional merge technique always select the vertex with the minimum distance, this scheme will affect the quality of final solution. Here we propose a novel merge operation which use the ranking-selection scheme. Assume that the shape boundary $C = \{p_1, p_2, \ldots, p_N\}$ has been segmented into M arcs $\widehat{p_{t_1}p_{t_2}}, \ldots, \widehat{p_{t_{M-1}}p_{t_M}}, \widehat{p_{t_M}p_{t_1}}$. Let $Z = \{p_{t_1}, p_{t_2}, \ldots, p_{t_M}\}$. For each point $p_{t_i} \in Z$, calculate the distance $q(p_{t_i}) = d(p_{t_i}, \overline{p_{t_{i-1}}p_{t_{i+1}}})$. The set W is then sorted into an ordered set $\{p_{k_1}, p_{k_2}, \ldots, p_{k_M}\}$ such that $q(p_{k_1}) \leq q(p_{k_2}) \leq \ldots \leq q(p_{k_M})$. The selection probability $\rho(p_{k_i})$ for point p_{k_i} is defined as

$$\rho(p_{k_i}) = \frac{i^{-r}}{\sum\limits_{j=1}^{M} j^{-r}}, \tag{5}$$



(a) chromosome (b) semicircle (c) leaf

Fig. 3. Three benchmark curves

where r is the parameter used to adjust the probability distribution. According to the selection probability, choose a point p_{k_i} from the set Z and merge the two adjacent arcs into a single one.

5.3 Algorithm Flow

The proposed algorithm has two parameters, one is the parameter of adjusting the probability distribution r, the other is the iteration number G.

input. A shape boundary $C = \{p_1, p_2, \ldots, p_N\}$ and a pre-specified tolerance error ε.

output. polygon B and its number of sides.

step 1. Randomly select two points from C and segment the boundary into two sections.

step 2. Repeat do split operation with ranking-selection to the boundary until the approximation error of the obtained polygon V is smaller or equal to ε. $V \to B$ and $0 \to k$.

step 3. Repeat do merge operation with ranking-selection to the boundary until the approximation error of the obtained polygon V is larger than ε.

step 4. Repeat do split operation with ranking-selection to the boundary until the approximation error of the obtained polygon V is smaller or equal to ε.

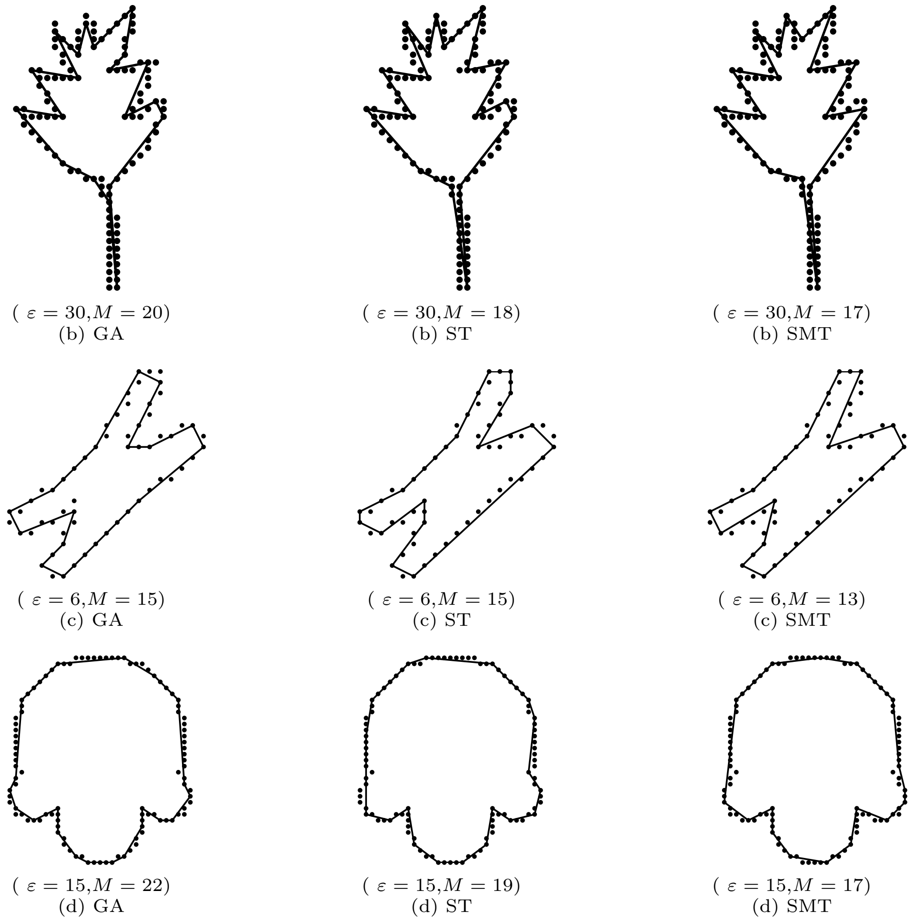
step 5. If the number of the sides of the polygon V is smaller than the number of the sides of the polygon B, then $V \to B$.

step 6. $k + 1 \to k$ and if $k \leq G$, then goto step 3.

step 7. output polygon B and its number of sides.

6 Experimental Results and Discussions

Three benchmarks, as shown in Fig. 3, are used to evaluate the performance of the proposed split and merge technique (SMT). Among them, (a) is a chromosome shape, (b) is a shape with four semi-circles and (c) is a leaf shape. The number of their boundary points is 60, 102 and 120 respectively. Their chain codes can be obtained from [9].



($\varepsilon = 30$,$M = 20$)
(b) GA

($\varepsilon = 30$,$M = 18$)
(b) ST

($\varepsilon = 30$,$M = 17$)
(b) SMT

($\varepsilon = 6$,$M = 15$)
(c) GA

($\varepsilon = 6$,$M = 15$)
(c) ST

($\varepsilon = 6$,$M = 13$)
(c) SMT

($\varepsilon = 15$,$M = 22$)
(d) GA

($\varepsilon = 15$,$M = 19$)
(d) ST

($\varepsilon = 15$,$M = 17$)
(d) SMT

Fig. 4. The comparative results of ST, GA and SMT, where ε is the specified tolerance error, M is the number of sides of the obtained approximating polygon

Two other methods, split technique (ST)[4] and Genetic algorithms (GA) [6], are used as comparisons with the proposed method. Each competitive methods are implemented on a PC with a PM 1.5 CPU under Windows XP. The parameter of SMT is set as : $G = 1500$ and $r = 1.8$. For shape boundary and a specified error tolerance ε, the simulation conducts ten independent runs. The best solution, average solution and variance of solutions during ten independent runs are listed in Table 1. Parts of best simulation results of three methods are shown in Fig. 4, where ε is the specified error tolerance and M is the number of sides of obtained approximating polygon.

From Table 1 and Fig. 4, we can see that, for the same tolerance error, SMT yields approximating polygon with relatively smaller number of sides than GA and ST. Variance is used to evaluate the stability of the three methods. Table 1

Table 1. Experimental results for ST, GA and SMT

Curves	ε	BEST			AVERAGE			VARIANCE		
		ST	GA	SMT	ST	GA	SMT	ST	GA	SMT
Leaf ($N = 120$)	150	12	15	11	14.4	15.4	12.3	2.3	0.5	0.7
	100	14	16	13	16.7	16.2	14.3	4.2	0.3	0.5
	90	15	17	13	17.7	17.4	14.3	1.8	0.4	0.7
	30	18	20	17	19.3	20.3	18.0	1.3	0.3	0.7
	15	21	23	21	24.5	23.1	22.2	4.5	0.4	0.4
Chromo-some ($N = 60$)	30	9	7	7	10.5	7.6	7.2	0.7	0.2	0.2
	20	10	8	7	10.7	9.1	8.0	1.1	0.3	0.2
	10	13	10	10	13.4	10.4	10.9	0.5	0.4	0.3
	8	13	12	11	14.0	12.4	11.9	0.4	0.3	0.5
	6	15	15	13	15.6	15.4	13.5	0.5	0.4	0.5
Semicirle ($N = 102$)	60	11	12	10	12.0	13.3	10.7	0.4	0.3	0.5
	30	14	13	13	15.1	13.6	13.9	1.7	0.4	0.3
	25	14	15	14	15.4	16.3	15.0	3.6	0.5	0.6
	20	15	19	15	18.1	19.5	17.0	6.5	0.3	1.1
	15	19	22	17	20.4	23.0	19.1	0.9	0.7	0.7

also shows that the proposed method SMT and genetic algorithms are more stable than ST. From all the experimental results, we can see that SMT outperforms genetic algorithms and the traditional split technique.

7 Conclusions

We have proposed a novel split and merge technique for solving error-bounded polygonal approximation. Since the ranking-selection scheme is used to choose split and merge points, the proposed method is not sensitivity to the initial segmentation against the shape boundary. The experimental results show that our method is superior to the traditional split technique and genetic algorithms.

Acknowledgement

The research work in this paper is partially sponsored by Shanghai Leading Academic Discipline Project, T0603.

References

1. Rafael, C.Gonzalez., Richard, E.Woods.: Digital Image Processing. Prentice Hall. (2002) 648-649
2. Sklansky, J., Gonzalez. V.: Fast Polygonal Approximation of Digitized Curves. Pattern Recognition. **12** (1980) 327–331
3. Ray, B.K., Ray, K.S.: Determination of Optimal Polygon from Digital Curve Using L_1 Norm. Pattern Recognition. **26** (1993) 505–509

4. Ramer, U.: An Iterative Procedure for the Polygonal Approximation of Plane Curves. Comput. Graph. Image. Process. **1** (1972) 244–256
5. Pikaz, A., Dinstein I.: An Algorithm for Polygonal Approximation Based in Iterative Point Elimination. Pattern Recognition Letters. **16(6)** (1995) 557–563
6. Yin, P.Y.: Genetic Algorithms for Polygonal Approximation of Digital Curves. Int. J. Pattern Recognition Artif. Intell. **13** (1999) 1–22
7. Goldberge, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley. Reading. MA. (1989)
8. Baker, J.F.: Adaptive Selection Methods for Genetic Algorithms. Grefenstette J J(Ed.).Proc. of the 1st Int'1. Conf. on Genetic Algorithms. Lawrence Earlbaum Associates, Hilladale, NJ. (1985) 101–111
9. Teh, H.C., Chin, R.T.: On Detection of Dominant Points on Digital Curves. IEEE Trans Pattern Anal Mach Intell. **11(8)** (1989) 859–872

Fast and Adaptive Low-Pass Whitening Filters for Natural Images*

Ling-Zhi Liao, Si-Wei Luo, Mei Tian, and Lian-Wei Zhao

School of Computer and Information Technology, Beijing Jiaotong University,
Beijing, 100044, China
sophiallz@163.com, swluo@center.njtu.edu.cn,
{tmlily, lw_zhao}@126.com

Abstract. A fast and simple solution was suggested to reduce the inter-pixels correlations in natural images, of which the power spectra roughly fell off with the increasing spatial frequency f according to a power law; but the $1/f$ exponent, α, was different from image to image. The essential of the proposed method was to flatten the decreasing power spectrum of each image by using an adaptive low-pass and whitening filter. The act of low-pass filtering was just to reduce the effects of noise usually took place in the high frequencies. The act of whitening filtering was a special processing, which was to attenuate the low frequencies and boost the high frequencies so as to yield a roughly flat power spectrum across all spatial frequencies. The suggested method was computationally more economical than the geometric covariance matrix based PCA method. Meanwhile, the performance degradations accompanied with the computational economy improvement were fairly insignificant.

1 Introduction

In a task of image analysis, the inter-pixels correlations within each individual image will result in vast inequities in variance at different frequencies of the image [1]. That is, the information at some frequencies might swamp the equally useful information at other frequencies, which may be troublesome for gradient descent techniques [2] searching for structures in images, such as independent component analysis (ICA) [3,4,5] and sparse coding analysis (SCA) [6,7,8].

The inter-pixels correlations within a natural image can be obtained by observation of its geometric covariance matrix [9]. The geometric covariance matrix is the covariance matrix of a data matrix, the columns of which are an image vector and its circular shifts. Since principle components analysis (PCA) [10,11] is an optimum transformation for decreasing pair-wise correlations between variables, it is possible to make use of PCA method to reduce the correlations between pair-wise pixels within one image. However, solving eigenvalues and corresponding eigenvectors from the geometric covariance matrix is always a non-trivial task. For example, if we want to

* Supported by the National Natural Science Foundation of China under Grant Nos. 60373029 and the National Research Foundation for the Doctoral Program of Higher Education of China under Grant Nos. 20050004001.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 343 – 352, 2006.

decrease the inter-pixels correlations in an image with 64×64 pixels in size, there will be 2^{12} random variables in the data matrix. It will then produce a geometric covariance matrix with 2^{24} entries, which is too big to compute. The PCA method is hereby limited when used to reduce inter-pixels correlations, even for an image of ordinary size.

Fortunately, the inter-pixels correlations within a natural image can also be measured by observation of its power spectrum [12] as well as its geometric covariance matrix. The link between the power spectrum and the geometric covariance matrix is the autocorrelation function. The autocorrelation function describes how closely related two pixels in an image are as a function of their relative separation. From one side, the autocorrelation function can be shifted to form each row of the geometric covariance matrix; the diagonal components of the geometric covariance matrix are equal to the autocorrelation function at zero separation. From the other side, the power spectrum and the autocorrelation function forms a Fourier pair. That is the Fourier transform (FT) of the autocorrelation function is the power spectrum, and the autocorrelation function is the inverse Fourier transform (IFT) of the power spectrum.

Under the translation invariance assumption [13] for natural images, any change of the power spectrum will also bring change to the autocorrelation function. Especially, when the power spectrum is a constant function, the autocorrelation function will be a delta function for sure, because the IFT of a constant function is a delta function. The geometric covariance matrix will then be a diagonal matrix. It suggests that when the power spectra of natural images are nearly flat, the inter-pixels correlations within the images are little. This is consistent with one of the hypotheses in neuroscience, that is the retina and the lateral geniculate nucleus (LGN) are dedicted to recording input visual information into a whitening form [14], for the power spectra of retina and LGN responses evoked by natural visual stimuli are essentially flat or white. It is obvious that the statistics of the visual environment have crucial influence on the way that the visual system process information [15,16].

In this article, we mainly discussed how to white natural images, the power spectra of which obeyed the following statistical rule: they fell with the spatial frequency f, according to a power law of $1/f^{\alpha}$, in which the exponent α was different from image to image. The essential of the proposed method was to filter each natural image with a combined whitening and low-pass filter. The whitening parameter, α_w, was adaptive to the current input image, not be fixed at the value of 2 as used in other works [1,17], so that the power spectrum of the whitened image is as flat as possible.

We begin by introducing the relationships among the autocorrelation function, the geometric covariance matrix, and the power spectrum. The next section describes the exciting properties of natural images by their power spectra, based on which we then derive a method for reducing the inter-pixels correlations within natural images. Performances of the geometric covariance matrix based PCA method and our power spectrum based method for whitening natural images are compared with each other for a 32×32 image. Further experimental results obtained by applying our method to natural images of larger size are described and analyzed afterwards. Finally, we make a conclusion and discuss experimental predictions that arise from the method.

2 Autocorrelation Function, Geometric Covariance Matrix and Power Spectrum

Let an N-by-M array, $I(x,y)$, denote the intensity function of some natural image with $N \times M$ pixels in size; $I(x+\Delta x_k, y+\Delta y_k)$ denote the result of circularly shifting $I(x,y)$ by Δx_k and Δy_k in the horizontal and vertical directions, respectively. The subscript k was an integer, ranging from 0 to $N \times M - 1$, and

$$\Delta x_k = \mathrm{mod}(k, N) \qquad \Delta y_k = \mathrm{div}(k, N) \tag{1}$$

where 'mod' and 'div' were used to dedicate computing the modulus and the quotient, respectively. It was obvious that there were, in total, $(N \times M)^2$ circular shifts for one natural image with $N \times M$ pixels in size.

The autocorrelation function was defined as

$$\begin{gathered} C(\Delta x, \Delta y, k) = \frac{1}{N \times M} \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} I(x+\Delta x_k, y+\Delta y_k) I(x+\Delta x_k+\Delta x, y+\Delta y_k+\Delta y) \\ \Delta x = 0,1,2,\cdots,N-1 \qquad \Delta y = 0,1,2,\cdots,M-1 \end{gathered} \tag{2}$$

Due to the stationarity [18] of natural image statistics, the autocorrelation function of $I(x,y)$ only depended on the relative separation between pixels, being independent of their absolute positions, therefore

$$C(\Delta x, \Delta y, 0) = C(\Delta x, \Delta y, 1) = \cdots = C(\Delta x, \Delta y, N \times M - 1) \tag{3}$$

Such a series of functions could be regarded as one function and simply represented by $C(\Delta x, \Delta y)$, with

$$C(\Delta x, \Delta y) = \frac{1}{N \times M} \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} I(x,y) I(x+\Delta x, y+\Delta y) \tag{4}$$

The function $C(\Delta x, \Delta y)$ was also called autocorrelation function, and it had the same dimensionality as the input image, that was $N \times M$.

Let I_k denote a column vector, in which the rows of $I(x+\Delta x_k, y+\Delta y_k)$ placed one after the other. Then a data matrix $\mathbf{I}$ could be organized by $\mathbf{I} = [I_0, \cdots, I_k, \cdots, I_{N \times M-1}]$. The geometric covariance matrix [9] was then $\mathbf{R} = \mathbf{I}^{\mathrm{T}}\mathbf{I} = \{R_{kl}\}$, with

$$\begin{aligned} R_{kl} &= \frac{1}{N \times M} \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} I(x+\Delta x_k, y+\Delta y_k) I(x+\Delta x_l, y+\Delta y_l) \\ &= C(|\Delta x_l - \Delta x_k|, |\Delta y_l - \Delta y_k|) \end{aligned} \tag{5}$$

Thus the autocorrelation function could be shifted to form each row of the geometric covariance matrix. Especially, when the autocorrelation function became a delta function only with non-zero at the zero separation, the geometric covariance matrix of the input image would be a diagonal matrix with zero everywhere except for the diagonal entries.

Let $S(u,v)$ denote the discrete Fourier transform (DFT) of the autocorrelation function $C(\Delta x, \Delta y)$, that was

$$S(u,v) = \frac{1}{N \times M} \sum_{\Delta y=0}^{M-1} \sum_{\Delta x=0}^{N-1} C(\Delta x, \Delta y) \exp\left[-j2\pi \left(\frac{u\Delta x}{N} + \frac{v\Delta y}{M} \right) \right] \qquad (6)$$

$$u = 0,1,2,\cdots,N-1 \qquad v = 0,1,2,\cdots,M-1$$

where u and v were the spatial frequency coordinates in the horizontal and vertical directions, respectively. An elementary but tedious computation could lead to

$$S(u,v) = \left| F(u,v) \right|^2 \qquad (7)$$

where $F(u,v)$ represented the DFT of $I(x,y)$. That meant the Fourier transform of the autocorrelation function was the power spectrum of the input image. Thus the autocorrelation function and the power spectrum formed a Fourier pair.

The translation invariance assumption [13] suggested that the intensity characteristics of natural images would not change even if the observing coordinates system changes from the space domain to the frequency one. The deep meaning of such an assumption for natural images, the alteration of the power spectrum in the frequency domain will also bring change to the autocorrelation function in the space domain. Especially, when the power spectrum was approximating a constant function, the autocorrelation function would be near a delta function, and then the geometric covariance matrix would be close to a diagonal matrix. In one word, when the power spectra of natural images were nearly flat, the inter-pixels correlations within the images would be very little.

3 Power Spectra Statistics for Natural Images

In this section we discussed the important properties of natural images by their power spectra. This topic was discussed in greater detail in other papers, such as [12]. However, since the conclusion of this section played an important part in the next section, it was discussed briefly here.

The power spectra were estimated for the ensemble of images of four different subjects, including natural scenes (from http://calphotos.berkeley.edu/), aerial images, man-made structures and faces (from http://sipi.usc.edu/database). All of the gray images were 512×512 pixels in size. Fig. 1 gave four sample images in our dataset, each of which was of different subjects.

The two-dimensional power spectra $S(u,v)$ of the four sample images were shown in Fig. 2, respectively. For the sake of the clarity, each 512×512 power spectrum was processed by averaging each 16×16 distinct region of the spectrum. The centers of such plots represented the low spatial frequencies. It could be seen that the power spectra of these images were quite characteristic, having greater values at low frequencies and decreasing sharply with the increasing frequency at all orientations.

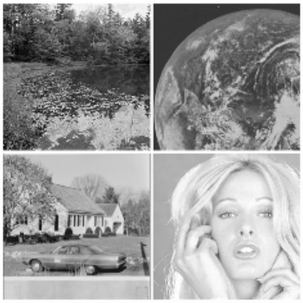


Fig. 1. Four sample images from the ensemble

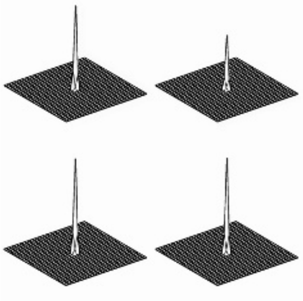


Fig. 2. Two-dimensional power spectra of the samples

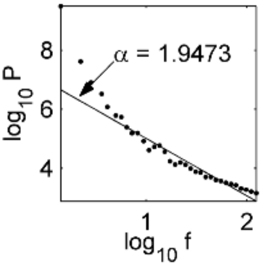


Fig. 3. Average power spectrum across all orientations

Fig. 3 gave the orientation averaged power spectrum, $P(f)$, of the natural scene from Fig. 1, on the log-log scale. The orientation power spectrum was calculated by

$$P(f) = \frac{1}{L_{\varphi}} \sum_{\varphi} S(f\cos\varphi, f\sin\varphi) \tag{8}$$

where $f = \sqrt{u^2 + v^2}$ was the spatial frequency and $\varphi = \arctan(u/v)$ was the orientation in the frequency polar coordinates system, respectively; L_{φ} was the number of orientations being computed. Note that the power at frequencies above 128 cycles per image were cut out from Fig. 3, for the highest frequencies were always easily corrupted by noise or affected by the effects of aliasing. The regression line in Fig. 3 was determined by linear curve-fitting of the average power spectrum plots. The value of 1.9473 for α was then negative to the slope of the regression line. It could be seen that the orientation averaged power spectrum approximately fell off with the spatial frequency according to a power law, as

$$P(f) \propto 1/f^{\alpha} \tag{9}$$

The average value of α for the four different subjects in our set of images were 2.1186, 2.3481, 2.5877 and 3.0223 respectively, in the order of natural scenes, aerial images, man-made structures and faces. The distribution of α for our images ensemble was nearly consistent with the conclusions in Ref. [12].

4 Adaptive Low-Pass and Whitening Filter

The prime difference between uncorrelated data and natural images by the power spectra was that uncorrelated data had its power uniformly distributed over the entire spectrum. In other words, the power spectrum of uncorrelated data was nearly as flat as a constant function, which was also testified by the decorrelating responses of retina and LGN evoked by natural visual stimuli [14]. Therefore, it would be feasible to white natural images simply by flattening their power-law power spectra into constant functions.

Since the orientation averaged power spectrum $P(f)$ of image $I(x,y)$ was nearly proportional to $1/f^{\alpha}$, the whitening filter for the image could be designed as

$$W(u,v)=\left(u^2+v^2\right)^{\frac{\alpha_w}{4}}\exp\left[-j2\pi\left(\frac{ux}{N}+\frac{vy}{M}\right)\right] \tag{10}$$

where α_w was named the whitening parameter. It could be seen that the whitening filter was, in fact, the result of multiplying a Fourier basis by the factor $\left(u^2+v^2\right)^{\alpha_w/4}$.

Fig. 4 (*left*) gave the real parts of a Fourier basis, when $N=16$ and $M=16$; Fig. 4 (*middle*) showed a corresponding whitening filter with $\alpha_w=2$. Fig. 4 (*right*) was a low-pass whitening filter, which would be discussed in detail later. In such plots, the centers stood for the low frequencies.

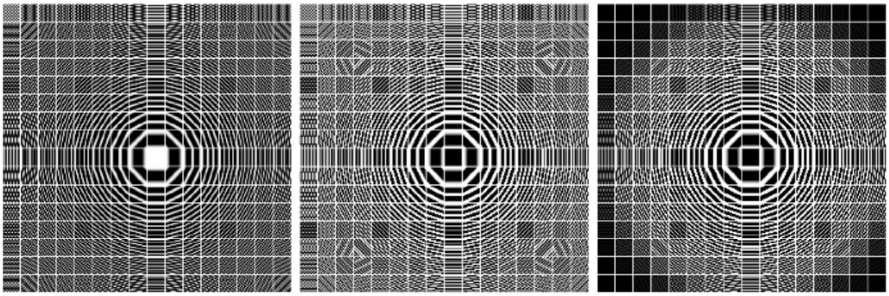


Fig. 4. Real parts of a Fourier basis (*Left*), a whitening filter (*Middle*) and a low-pass whitening filter (*Right*)

Filtering the image $I(x,y)$ with the whitening filter, we would obtain a series of frequency coefficients $F_w(u,v)$, with

$$F_w(u,v)=\left(u^2+v^2\right)^{\frac{\alpha_w}{4}}F(u,v) \tag{11}$$

Let $I_w(x,y)$ denote the IFT of $F_w(u,v)$, the two-dimensional power spectrum of $I_w(x,y)$ would be

$$S_w(u,v)=\left(u^2+v^2\right)^{\frac{\alpha_w}{2}}S(u,v) \tag{12}$$

The orientation power spectrum was then

$$P_w(f)=f^{\alpha_w}P(f)\propto f^{\alpha_w}/f^{\alpha} \tag{13}$$

It was obvious that when the whitening parameter α_w was equal to the $1/f$ exponent, α , of the original image $I(x,y)$, the orientation averaged power spectrum of $I_w(x,y)$ would be nearly a constant function. Therefore, we could regard $I_w(x,y)$ as whitening form of the image $I(x,y)$.

Furthermore, to guarantee that no significant noise at the highest frequencies could be allowed to pass, the whitening filter in eq. (10) was then improved to be a combined low-pass and whitening one as

$$LW(u,v) = \left(u^2 + v^2\right)^{\frac{\alpha_w}{4}} L(u,v) \exp\left[-j2\pi\left(\frac{ux}{N} + \frac{vy}{M}\right)\right] \tag{14}$$

where $L(u,v)$ was an exponential low-pass filter as

$$L(u,v) = \exp\left[-\left(\frac{\sqrt{u^2 + v^2}}{f_c}\right)^n\right] \tag{15}$$

The frequency f_c in the low-pass filter was called cut-off frequency, at which the attenuation of frequency components was started; and n was the steepness parameter. The values for f_c and n were selected to guarantee that not only no significant noise could be allowed to pass, but also the total power of the original image could not be attenuated highly.

The coefficients by filtering $I(x,y)$ with the low-pass whitening filter would be

$$F_{lw}(u,v) = L(u,v) F_w(u,v) \tag{16}$$

If $I_{lw}(x,y)$ was the IFT of $F_{lw}(u,v)$, its orientation averaged power spectrum would be

$$P_{lw}(f) \approx P_w(f) \tag{17}$$

because the employed low-pass filter would not change or only change little the power of the original image at the low frequencies, which was the primary part of the total power. Fig. 4 (*right*) gave the real parts of a low-pass whitening filter, when $N = 16$, $M = 16$, $\alpha_w = 2$, $f_c = 6.4$ and $n = 4$.

In summary, the processing of whitening an image $I(x,y)$ could be summarized as:

1. Estimate the value for the $1/f$ exponent, α, from the orientation averaged power spectrum plots of $I(x,y)$;
2. Calculate the frequency coefficients $F_{lw}(u,v)$ by filtering $I(x,y)$ with the low-pass whitening filter $LW(u,v)$, in which the whitening parameter α_w is equal to α;
3. Compute the inverse Fourier transform $F_{lw}(u,v)$ to obtain the whitening form the original image $I(x,y)$ in the space domain, that was $I_{lw}(x,y)$.

5 Experiments and Results

A small image piece of 32×32 pixels was first taken on trial to compare the performance of the method proposed in this article with that of the geometric covariance matrix based PCA method. The first 256 from the total 1024 principle components of the 32×32 image were shown in Fig. 5, with the components of high variances being shifted into the center. The geometric covariance matrices for natural images had repeated eigenvalues indeed. As illustrated in Fig. 5, the eigenvectors with the same eigenvalues had the same spatial frequency information, while there was a phase lag between them.

Fig. 6 (*top*) was the 32×32 image piece. Because of low resolution, it was blur indeed. Fig. 6 (*bottom left*) represented its two-dimensional PCA representation. And Fig. 6 (*bottom* right) was its whitened result by employing our proposed method. For the sake of clarity, all three two-dimensional images in Fig. 6 was specified so that the minimum value in the image displayed as black, the maximum value displayed as white, and values in between displayed as intermediate shades of gray. It was obvious that our whitened result kept the same edge or line structures as the original image, while it was very difficult or even impossible to make out the appearance of the original image from the decorrelating form by employing PCA method. The deep reason for this was that we only changed the amplitude spectra of natural images, without scrambling their phase spectra, which decided the higher-order statistics of natural images, such as edges or lines [18].

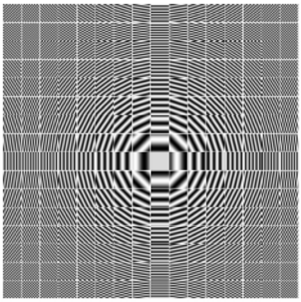


Fig. 5. First 256 principle components of the small image piece

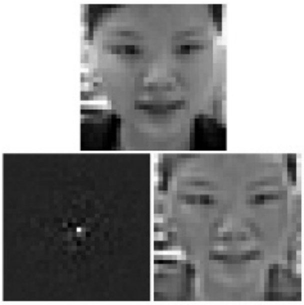


Fig. 6. Image piece and its decorrelating forms by PCA and our methods, respectivley

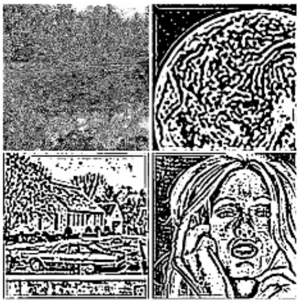


Fig. 7. Whitened results of four sample images

To measure the degree of inter-pixels correlations being reduced from the original image, we could use the measure η, which was

$$\eta = \frac{\left|\mathbf{C}_{\mathrm{X}} - \mathbf{C}_{\mathrm{Y}}\right|^2}{\left|\mathbf{C}_{\mathrm{X}} - \mathbf{D}\right|^2} \tag{18}$$

where $\mathbf{C}_{\mathrm{X}}$ was the normalized autocorrelation function of the raw image, in which the value at the zero separation was 1; $\mathbf{C}_{\mathrm{Y}}$ was the normalized autocorrelation function of the transformed image; and $\mathbf{D}$ was a sparse matrix with zero everywhere except for the zero separation position. Values of η near 100% indicated good performance; values of η near 0% indicated bad performance, then. Ref. [19] used another measure to weigh degree of correlations being reduced, but it needed to calculate the geometric covariance matrix of the input image, which was easily beyond the reasonable computation when the image was large, as mentioned above.

For the 32×32 image piece from Fig. 6, the values for η when using PCA method and ours were 97.97% and 91.55%, respectively, as illustrated in Table 1. In addition, the time needed to calculate the PCA decorrelating form was about 24.1961 second, while it only spent 0.0047 second to compute our whitening form, because we employed the well-known fast Fourier transform algorithm. Thus it could be seen that although the performance of the geometric covariance matrix based PCA method was better than our power spectrum based method, the related operation speed was pretty slower indeed. Moreover, when the input image became larger such as 64×64, the PCA method could not work anaymore, for the geometric covariance matrix would be as large as 2^{24}, beyond the reasonable computation.

Table 1. Performace Comparison between PCA Method and Our Method

	PCA method	Our method
η	97.97%	91.55%
Time consuming	$24.1961'$	$0.0047'$

Fig. 7 gave the whitened results of the four 512×512 sample images from Fig. 1, which also had the same edges or lines as the original images. The values for η were then listed in Table 2, when α_w was adaptive to the input image and fixed at the value of 2 [1,17], respectively. It was obvious that the whitening filters with adaptive α_w performed a little more well than the fixed ones. Meanwhile, the time needed to whiten each 512×512 natural image was about 1.3650 second.

Table 2. Performance of Whitening Filters for Four Sample Images with Adaptive and Fixed α, respectively

	Top left	Top right	Bottom left	Bottom right
$\alpha_w=\alpha$	99.44%	99.15%	98.82%	98.78%
$\alpha_w=2$	97.89%	97.42%	97.74%	97.89%

6 Discussion

We have investigated that the second-order redundancy in natural images would be decreased if the first-order redundancy of the power spectra were reduced in the frequency domain. By making use of fast Fourier transform algorithms, our method was computation saving and therefore could be used to whiten natural images of large size. Therefore, in the sparse and over-complete models for natural images, we could whiten each image into a decorrelating form before dividing them into small patches, to keep away from the limitation that the number of basis functions should not be larger than the dimensionality of the input data, which was necessary for PCA method, on the contrary.

References

1. Olshausen, B.A., Field, D.J.: Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? Visual Research 37 (1997) 11–25
2. Amari, S.: Natural Gradient Works Efficiently in Learning. Neural Computation 10 (1998) 251–276
3. Bell, A., Sejnowski, T.: The Independent Components of Natural Scenes Are Edge Filters. Vision Research 37 (1997) 3327–3338
4. van Hateren, J.H., van der Schaaf, A.: Independent Component Filters of Natural Images Compared with Simple Cells in Primary Visual Cortex. Proc. Royal Society ser. B, 265 (1998) 359–366
5. Hyvärinen, A., Oja, E.: Independent Component Analysis: Algorithms and Applications. Neural Networks 13 (2000) 411–430
6. Olshausen, B.A., Field, D.J.: Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images. Nature 381 (1996) 607–609
7. Olshausen, B.A.: Principles of Image Representation in Visual Cortex. In: Chalupa, L.M., Werner, J.S. (eds.): The Visual Neurosciences. MIT Press, Cambridge Mass, USA (2003)
8. Lewicki, M.S., Sejnowski, T.J.: Learning Overcomplete Representations. Neural Computation 12 (2000) 337–365
9. Camelio, J.A., Hu, S.J., Marin, S.P.: Compliant Assembly Variation Analysis Using Component Geometric Covariance. Journal of Manufacturing Science and Engineering, 126 (2004) 355-360
10. Jolliffe, I.: Principal Component Analysis. Springer-Verlag, Berlin Heidelberg New York (1986)
11. Basilevsky, A.: Statistical Factor Analysis and Related Methods: Theory and Applications. Wiley, New York (1994)
12. Schaaf, A., Hateren, J.H.: Modelling the Power Spectrum of Natural Images: Statistics and Information. Vision Research 36 (1996) 2759–2770
13. Simoncelli, E.P., Olshausen, B.A.: Natrual Image Statistics and Neural Representation. NeuroSecience, Annual Review 24 (2001) 1193–1216
14. Dan, Y., Atick, J.J., Reid, R.C.: Efficient Coding of Natural Scenes in the Lateral Geniculate Nucleus: Experimental Test of a Computational Theory. Neuroscience 16 (1996) 3351–3362
15. Field, D.J.: Relations between the Statistics of Natural Images and the Response Properties of Cortical Cells. Optical Society of America 4 (1987) 2379–2394
16. Olshausen , B.A.: How Close We Are to Understand V1? Neural Computation 17 (2005) 1665–1699
17. Atick, J.J., Li, Z. Redlich, A.N.: Understanding Retinal Color Coding from First Principles. Neural computation 1 (1992) 559–572
18. Field, D.J.: What Is the Goal of Sensory Coding? Neural Computation 6 (1994) 559–601
19. Massih, H., Pearl, J.: Comparison of the Cosine and Fourier Transform of Markov-1 Signals. IEEE Trans. on Acoustics, Speech, and Signal Processing (1976) 428–429

An Exhaustive Employment of Neural Networks to Search the Better Configuration of Magnetic Signals in ITER Machine

Matteo Cacciola, Antonino Greco,
Francesco Carlo Morabito, and Mario Versaci

University Mediterranea of Reggio Calabria, Department of Informatics,
Mathematics, Electronics and Transportation (DIMET) 89100 Reggio Calabria, Italy
{matteo.cacciola, antonino.greco, morabito}@unirc.it,
versaci@ing.unirc.it
http://neurolab.ing.unirc.it

Abstract. Concerning the control of plasma column evolution in ITER machine, the reconstruction of the plasma shape in the vacuum vessel represents an important step. In this work, starting from magnetic measurements, a soft computing approach to estimate the distances of the plasma boundary from the first wall of the vacuum vessel is carried out by means of Neural Networks (NNs). In particular, Multi-Layer Perceptron (MLP) nets have been exploited for the purpose. Finally, to verify the robustness of the proposed approach, any different database and number of input parameters has been used.

1 Introduction

In order to control the plasma column evolution within the vacuum chamber of the ITER machine [1], it is required to timely identify the position and shape of the plasma. Regarding ITER configuration, the plasma shape is delineated bye some distances calculated from the plasma boundary to the first wall in the vacuum vessel (gaps) (Fig. 1). Then, the problem under study is an inverse problem in which the input of the final identification device is the set of the simulated magnetic measurements and the output are the gaps. These solutions are called magnetohydrodinamic equilibria (MHD model). The aim of our work is to supply a contribution to the reconstruction of the plasma and the position (outputs parameters), using flux (Ψ) and field (B) measurements (inputs parameters). Practically, an inverse approach to solve the reconstruction problem has been followed. So that, having N input data related to N-functionals of flux (magnetic signals), we reconstruct $\Psi(R, Z)$, describing the equilibriums surface ($\Psi(R, Z) = \cos(t)$), and the current profile $J(R, Z)$, when both the plasma current and the distance between plasma-first wall (Gap) are known and the Grad-Shafranovs equation [1], [2] is satisfied. In this case, when small perturbations take place, big variations of the solution occur. A configuration that does not take into account the problem above mentioned cannot be considered as an

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 353–360, 2006.

optimal one: particularly, even if just a few output parameters are affected by errors, the reconstruction of plasma shape is not correct. Then, we need different magnetic measurements carried out by a lot of external probes with respect to the first wall. In order to improve the goodness of results, we also exploit internal measurements. In the recent years, the neural computing approach has emerged as a successful framework for fast analysis of multi-channel data in plasma shape recognition [1], [2], [3]. The similar procedure of plasma shape reconstruction's has been made by means the mixed approach Functional Parameterizations and Principal Component Analysis [4]. The NN yields flexibility: indeed, the identification model (yet non linear, sigmoidal, with respect to linear combinations of the measurements) is dependent on non-orthogonal input combinations, can retain some information possibly included in the minor components (for example present in the transition between two different plasma shape configurations, like X-point and limiter ones), is less prone to noise in the measurements and permits simple CAD procedures on the architecture of the model. This is very useful in the hardware implementation of the processors. In this paper, MLPs have been exploited to solve the inverse problem reconstructing the plasma shape in the vacuum vessel in ITER machine. The paper is organized as follows: in section 2, a brief description of numerical database is presented; section 3 describes the NN-based approach; some important results take place in section 4. Finally some conclusions are drawn.

2 The Exploited Numerical Database

Using the ITER coil and vessel geometry [4], including the 6 dominant passive current eigenmodes, a database of 4848 lower single null equilibria has been generated by the Plasma Data Analysis Group (PDAG), Physics Department, University College Cork, Association EURATOM-DCU. The equilibria were generated using a Database Generation and Analysis Package (DGAP) which has been developed by PDAG. The core equilibrium calculation in DGAP is performed by the Garching Equilibrium Code (GEC). The database consists of Ip

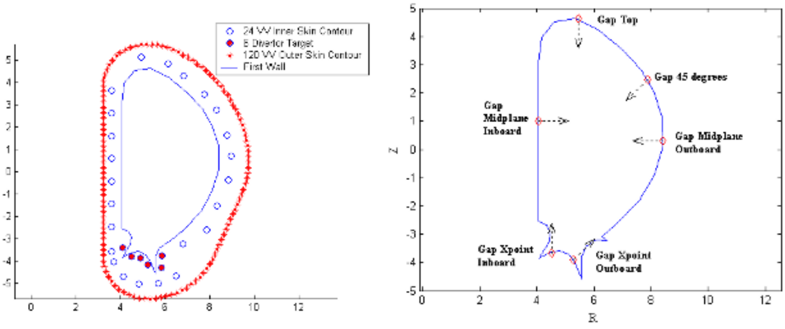


Fig. 1. Magnetic signals dislocations (left) and gap position (right)

Table 1. Structure of exploited numerical database

6 PF (Poloidal Field) 6 CS (Central Solenoid) 6 Mode Structure	Parameters associate to the currents in the windings and the passive currents presents in the structure
24 B_Tangential Signals on the Vacuum Vessel Inner Skin Contour 24 B_Normal signals on the Vacuum vessel Inner Skin Contour 6 B_Tangential signals below the Divertor Contour 6 B_Normal signals below the Divertor Contour 120 B_Tangential signals on the Vacuum Vessel Outer Skin Contour 120 B_Normal signals on the Vacuum Vessel Outer Skin Contour	Parameters associate to the magnetic signals and measured by means of magnetic probes
71 Plasma Parameters	Parameters associate to geometry and structure of the plasma

= 15MA, Bo = 5.3T (at R = 6.2 m) lower X-point plasmas The coils were modeled by partitioning the rectangular poloidal cross-section for each coil into sub-regions, each of which contains a single computational winding located at the centroid of the sub-region and which corresponds to approximately 10 physical windings (this number varies from coil to coil).

The database structure is presented in Table 1 in which a brief description of each parameter takes place. We have in the whole dataset 389 variables and 4848 samples (equilibria). The process of merging features is explicitly carried out in NNs approaches by means of learning process. In NNs the output of hidden layers of neurons build an internal representation of the problem. To be useful such intermediate representation of the data must preserve distance among patterns. This means that similar patterns must be represented by similar feature vectors. The data are thus clustered around specific classes of patterns. This is precisely what we are asked for in our identification problem. This implies a first derivative discontinuity of the mapping plasma parameter-magnetic measurement. It is a fact that numerical global regression, i.e. regressions carried out on the whole database which includes plasmas from all of the six possible categories, suffer from inadequacy. Global regressions typically show an error level about twice that the least accurate individual category regression. This inability is mainly related to the handling of the first order discontinuities in parameter behavior across category transitions.

In the Fig. 1 we reported the position of used magnetic probe and Gap location. The gap are a particular plasma parameters that represented the distance between plasma-first wall. With the same characteristic the used database are three; the difference is the noise level added in the simulation phase. The used value are: 2mTesla, 10mTesla and 20mTesla [4]; we emphasize that the added signal noise concerning the tangential and normal magnetic field.

3 NNs: An Overview

Artificial Neural Network (ANN) implements a non linear function mapping one multidimensional space, $\{\mathbf{x}\}$, into another one, $\{\mathbf{z}\}$ [3]. This function has a predefined structure but contains several parameters which are going to be determined during the training phase which consists in the evaluation of the parameters which minimize the differences between the target output t and the network output, z. Among several possible structures of the network, we use a so called, feed-forward MLP. This kind of network is known to approximate arbitrarily any continuous multi dimensional mapping [5]. The h-th component of the output vector $(h = 1, ..., n_z)$, can be written as:

$$z_h = F(\sum_{i=1}^{n_y} \mathbf{WY}_{hi} y_i) \cdot y_i = F(\sum_{j=1}^{n_x} \mathbf{WX}_{ij} x_j). \tag{1}$$

where: y_i is the i-th component of the output of the first layer; z_h is the h-th component of the networks output; $\mathbf{W}$ is the vector of link weights; n_x, n_y and n_z are the dimension of the input vector, the number of the hidden neurons and the dimension of the networks output respectively; F is a non-linear function [3]. In each layer, the input variable to the specific layer is transformed first linearly, by means of a matrix ($\mathbf{WX}$ and $\mathbf{WY}$ for the first and the second layer respectively) and then by a non-linear function. The values of the $(n_x * n_y + n_y * n_z)$ unknown elements of the matrixes $\mathbf{WX}$ and $\mathbf{WY}$ are found by minimizing an error function of the type:

$$E = 0.5 * \sum_{k=1}^{n} [\mathbf{z}(\mathbf{x}_{(k)}, \mathbf{WX}, \mathbf{WY} - \mathbf{t}_{(k)})]^2. \tag{2}$$

in which the sum is extended to the whole training set. A slow but reliable method to minimize the above equation is known as back-propagation algorithm [6] and consist of evaluating the derivatives of E with respect to the elements of the $\mathbf{WX}$ and $\mathbf{WY}$ matrixes and correct the unknown parameters using gradient descendent in the following way:

$$\mathbf{WX}_{ij}^{n+1} - \mathbf{WX}_{ij}^{n} = -\delta \frac{\partial E}{\partial \mathbf{WX}_{ij}}. \tag{3}$$

where δ is an appropriate learning rate parameter and n is the iteration number.

4 NNs Approach: Reconstruction Results

In this Section, we shall report about the performance of the NNs approach in the test cases already introduced in the previous Section. The NNs models used are based on a scheme with two hidden layers, the first one of linear transfer function, acting as a redundancy reduction layer, the second one with hyperbolic tangent transfer functions, that is responsible for the non-linearity of the resulting model; the number of nodes in the NN is shown for each column. A typical example of NN is reported in the Figure 2.

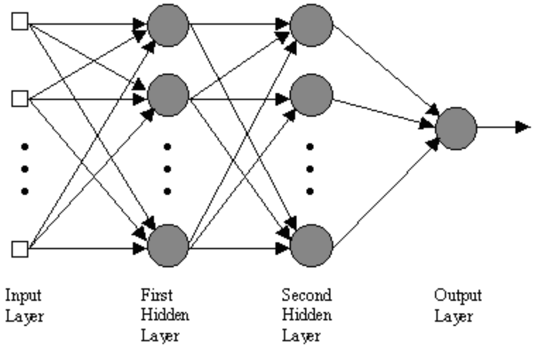


Fig. 2. The general representation of MLP network

4.1 Outer Skin Vessel Measurements: Sensitivity to Sensor Sub-sets Failure

The first comparative analysis has been conducted with respect to a diminution of the number of sensors; the committed error is evaluated by means of Root Mean Square Error (RMSE). Table 2 reports the obtained results in which, in the first line for both, the used configuration is reported; 240-30-40-6 is a network that has the input layer of 240 neurons two hidden layer of 30 and 40 neurons respectively, and an output layer of 6 neurons (6 gaps, in our case).

4.2 Inner Skin Vessel Measurements

Sensitivity to Noise Model. In according to previous notation, the exploited NN is 60-16-25-6, where 60 are the Inner Skin B measurements. RMSE is evaluated with respect to the added noise (Table 3).

Table 2. Summary of best results

	GapXin	GapXo	GapMo	Gap45	GapTop	GapMin
240-30-40-6	0.0347	0.0275	0.0223	0.0235	0.0545	0.0224
120-25-35-6	0.0395	0.0291	0.0265	0.0285	0.0565	0.0288
80-25-35-6	0.0415	0.0315	0.0277	0.0301	0.0601	0.0261
60-20-30-6	0.0455	0.0355	0.0307	0.0362	0.0680	0.0332
48-18-25-6	0.0465	0.0386	0.0315	0.0367	0.0677	0.0343
40-10-20-6	0.0584	0.0406	0.0342	0.0401	0.0822	0.0355
20-8-15-6	0.0622	0.0455	0.0397	0.0422	0.0855	0.0441

Table 3. Best results with respect to added signal noise

	2 mT + 2 mT	10 mT + 10 mT	20 mT + 20 mT
GapXin	0.0088	0.0128	0.0177
GapXo	0.0077	0.0111	0.0161
GapMo	0.0089	0.0155	0.0203
Gap45	0.0095	0.0158	0.0199
GapTop	0.0298	0.0354	0.0412
GapMin	0.0066	0.0122	0.0197

Table 4. Best results with respect to diminution of Inner Skin measurements

	60	30	20
GapXin	0.0128	0.0148	0.0221
GapXo	0.0111	0.0151	0.0187
GapMo	0.0155	0.0191	0.0356
Gap45	0.0158	0.0193	0.0287
GapTop	0.0354	0.0378	0.0514
GapMin	0.0122	0.0134	0.0316

Table 5. Best results respect to full Inner Skin measurements and absence of Divertor module

	60	30
GapXin	0.0128	0.0532
GapXo	0.0111	0.0527
GapMo	0.0155	0.0187
Gap45	0.0158	0.0173
GapTop	0.0354	0.0401
GapMin	0.0122	0.0152

Sensitivity to sensor sub-sets failure. In Table 4 we report the increasing of error when we consider a recursive diminution of the input parameters.

Sensitivity to sensor sub-sets failure. Table 5 is referred to the failure mode analysis for the divertor coils, that are fed out through a common connector. The Full input (60 Inner Skin B measurements) and absence of Divertor Module Measurements (48 inputs) are used.

Figs. 3 and 4 resume the results achieved with the NN approach. In particular, Fig. 3 reports, on the left hand, the reconstruction accuracy for the 6 gaps by using only external (outer skin) measurements (7 blocks of measurements have been used). Fig. 3 (at right) reports the results achieved by using the Inner Skin Vessel measurements for three different levels of noise. On the left of Fig. 4 the sensitivity of the reconstruction at the failure of sensors (60, 30 and 20 sensors)

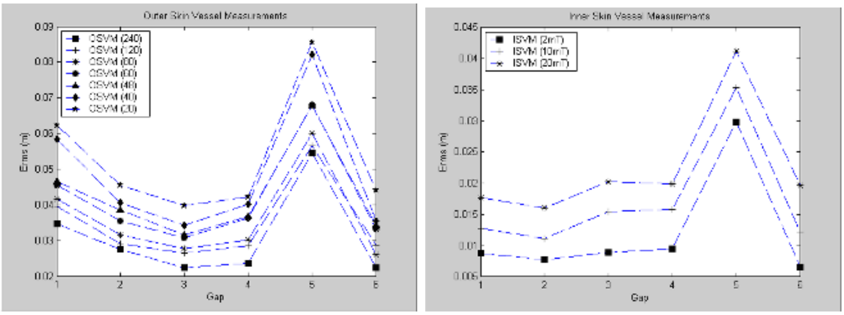


Fig. 3. Reconstruction accuracy for the six gaps using outer (left) and inner (right) skin vessel measurements. In particular, it is shown: sensitivity to the noise added in the database of equilibria at left; sensitivity to the failure of subsets of sensors at right.

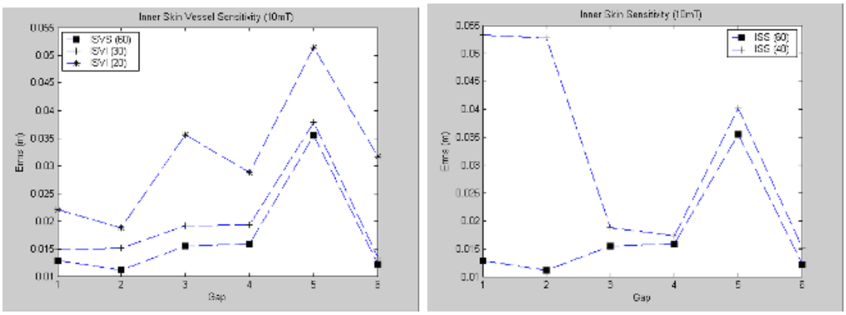


Fig. 4. Reconstruction accuracy for the six gaps using inner skin vessel measurements. Sensitivity to the failure of: some subsets of sensors (left); the subset of sensors in the divertor cassette (right).

is reported. Right part of Fig. 4 refers to the failure mode analysis in the case of lack of divertor sensors.

5 Conclusions

The paper has presented the main results of the analysis carried out on noisy data-base based on ITER configuration. Stochastic approaches have been exploited throughout the report. The obtained results have shown that NNs can be useful in order to reconstruct plasma shape in ITER configuration. The distributed representation can be very useful to face possible failure in the sensors, while the extrapolation capabilities are comparable with other techniques. The global performance can benefit from optimization of the input signals. The large redundancy of the input vector is an important issue to be investigated in future analysis, also to cope with the difficulty of learning in a multi-dimensional space with a large number of collinear and/or seemingly useless inputs. Others advantages of the MLP networks are the continuity of the solution and their employment in a multidimensional space having a great dimension.

References

1. Wesson, J.: Tokamaks. Oxford Science Pub. (1987).
2. Matsukawa, M. et al.: Application of Regression Analysis to Deriving Measurements Formulas for Feedback Control of Plasma Shape in JT-60. Plasma Physics and Controlled Fusion, Vol. 34, No. 6 (1992) 907-921.
3. Bishop, C. M.: Neural Network for Pattern Recognition. Clarendon Press, Oxford (1995).
4. Morabito, F.C. et al.: Final report on EFDA Study Contract FU05 CT 2002-00162 (EFDA 02-1001).
5. Morabito, F.C., et al.: On Line Plasma Shape Identification in a Tokamak Reactor Via Neural Network. Proc. of V Italian Workshop on Neural Networks, Word Scientific Publishing (1992) p. 349.
6. Morabito, F.C., Versaci, M.: A Fuzzy-Neural Approach to Real Time Plasma Boundary Reconstruction in Tokamak Reactors. IEEE ICNN International Conference on Neural Networks, Houston, Texas (1997) pp. 43-47.

Ultra-Fast fMRI Imaging with High-Fidelity Activation Map

Neelam Sinha[1], Manojkumar Saranathan[1], A.G. Ramakrishnan[1],
Juan Zhou[2], and Jagath C. Rajapakse[2]

[1] Indian Institute of Science, Bangalore, India
{neelam, manojk, ramkiag}@ee.iisc.ernet.in
[2] NTU, Singapore
{zhou0025, asjagath}@ntu.edu.sg

Abstract. Functional Magnetic Resonance Imaging (fMRI) requires ultra-fast imaging in order to capture the on-going spatio-temporal dynamics of the cognitive task. We make use of correlations in both k-space and time, and thereby reconstruct the time series by acquiring only a fraction of the data, using an improved form of the well-known dynamic imaging technique k-t BLAST (Broad-use Linear Acquisition Speed-up Technique). k-t BLAST (k-tB) works by unwrapping the aliased Fourier conjugate space of k-t (y-f space). The unwrapping process makes use of an estimate of the true y-f space, obtained by acquiring a blurred unaliased version. In this paper, we propose two changes to the existing algorithm. Firstly, we improve the map estimate using generalized series reconstruction. The second change is to incorporate phase constraints from the training map. The proposed technique is compared with existing k-tB on visual stimulation fMRI data obtained on 5 volunteers. Results show that the proposed changes lead to gain in temporal resolution by as much as a factor of 6. Performance evaluation is carried out by comparing activation maps obtained using reconstructed images, against that obtained from the true images. We observe upto 10dB improvement in PSNR of activation maps. Besides, RMSE reduction on fMRI images, of about 10% averaged over the entire time series, with a peak improvement of 35% compared to the existing k-tB, averaged over 5 data sets, is also observed.

1 Introduction

Magnetic resonance imaging (MRI) has emerged as a powerful tool in medical imaging and diagnosis in the last decade, due to its non-invasive nature and excellent soft-tissue contrast. Although high spatial resolution images are essential in medical diagnosis and image analysis, high temporal resolution is critical in applications like dynamic contrast-enhanced MRI or functional MRI (fMRI), where dynamic events are monitored. Today, fMRI has the potential to probe neurophysiological activation in the brain at a much higher spatial resolution than that offered by other non-invasive neuroimaging techniques like PET. The

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 361–368, 2006.

high sensitivity measurement of blood oxygenation level dependent (BOLD) signal modulation points to regions in the cortex responsible for the underlying activity. Currently fMRI applications interrogate neural activity changes only on the order of seconds, although neural activity happens on time scales of the order of milliseconds. One way of increasing the temporal resolution is to reconstruct high quality images from partial data. Parallel imaging methods can also be used to achieve accelerated imaging, but they require customized hardware. Parallel imaging involves utilizing an array of receiver coils with varying coil sensitivities, instead of a single coil, with homogeneous sensitivity.

In MRI, data is sampled in the spatial frequency domain of the object being imaged (called k space), directly leading to the well known trade-off between temporal and spatial resolution. Parallel imaging techniques [1,2] are gaining popularity but they require customized hardware. However, partial data-based reconstruction have no such requirements. Partial data acquisition involves acquiring a pre-determined region in k-space. Many techniques like Keyhole [3], Reduced encoding Imaging by Generalized series Reconstruction (RIGR) [4], Two-reference Reduced encoding Imaging by Generalized series Reconstruction (TRIGR) [5], Unaliasing by Fourier-Encoding the Overlaps Using the Temporal Dimension (UNFOLD) [6], k-t BLAST [7], that reconstruct images from partial data, have been reported. Some of the methods use direct replacement, while others extrapolate the missing values using correlations in k-space and/or time. Keyhole is the simplest known technique where during the course of dynamic changes only low frequencies in k-space are acquired, while the unacquired high frequencies are simply replaced by the corresponding values obtained from a static high-resolution acquisition. However, discontinuities in reconstructed k-space lead to artifacts in images, and hence higher acceleration factors cannot be explored. Methods like RIGR and TRIGR use generalized-series modeling to estimate the unacquired values in k-space. High resolution static images serve as estimates to obtain the corresponding values at instants of dynamic changes. Both these methods linearly fit the unacquired values in terms of the acquired data, and basically solve a system of linear equations. Reported works using these methods claim acceleration factors of 4-6. However, methods like UNFOLD and k-tB are radically different from the above. They employ a sparse acquisition scheme that results in a known form of aliasing that is eventually unwrapped either using temporal filtering (UNFOLD) or using a low resolution, alias-free training map (k-tB). In this paper, we propose changes to two aspects of the existing k-tB algorithm. Firstly, the estimates obtained from the training map are improved using generalized series modeling (labelled as RIGR in further references). The second proposed change is the incorporation of phase constraints obtained from the alias-free training map. These two changes together were utilized for reconstruction of fMRI data sets obtained from a photic stimulation experiment, and improvements in resulting images were quantified. The paper is organised as follows. The variations proposed to the existing k-tB technique is given in section 2. The data used and results obtained are discussed in section 3. Finally, section 4 concludes the paper.

2 Proposed Method

2.1 Data Acquisition Schemes

In the original k-tB scheme [7], the training and actual data acquisitions are done at disjoint instants of time, and follow different sampling schemes [8]. The training data samples only low-frequency k-space data, while the actual data acquisition is along a pre-designed sparsely sampled lattice, as shown in Fig. 1(a). A variation of data acquisition scheme that couples both the training and actual scans is shown in Fig. 1(b). This is a variable density sampling lattice. This scheme was chosen in order to minimize the mismatch between training and data scans. This scheme of acquisition reduces the acceleration factor achievable, but eliminates possible artifacts due to mis-registration. In our trials, we utilized this variable-density sampling scheme.

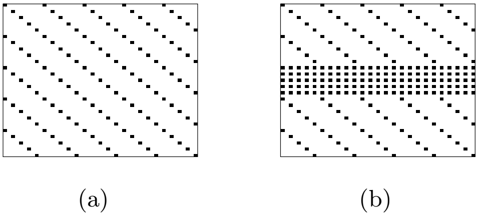


Fig. 1. Data acquisition (a) Uniform density (Existing) (b)Variable density (Utilized)

2.2 Training Map

The reported work of Hansen et al [9], deals with how the quality of training data influences the working of k-tB, in contexts where training and actual data are acquired at disjoint instants of time. It reports that increasing the number of time frames for which the training data is acquired, results in only a negligible decrease of reconstruction error. It also reports that filtering of the training data in order to reduce truncation artifacts had minor impact on reconstruction errors. However, in a variable-density acquisition scheme like ours, training data is available at all time frames of the experiment. We explored the impact of including higher frequencies in the training data, on the working of k-tB. We compared k-tB reconstructions that use low resolution training data against k-tB reconstructions that use all the frequencies (ideal training) in the training map. It is seen that the errors can be brought down using higher frequencies in the training map, by a factor of 2. The disparity in the two reconstructions led us to explore the possibility of obtaining an improved resolution training-map using the acquired low frequencies. It must be observed that at locations in the aliased y-f space, where the signal is dominated by noise, the values from the training map that are chosen as estimates, can lead to meaningful results only if the estimate is close to the truth.

2.3 Proposed Variations to *k*-*t*B

The proposed method generates an improved-resolution training map, despite acquiring only the lower spatial frequencies. This is done by extrapolation using the generalized series model, which requires one full-resolution acquisition. The high-resolution static acquisition serves to estimate the missing high-frequencies in the training map. The working of the generalized series modeling is outlined below.

Generalized series modeling: In generalized series modeling, the missing high spatial frequencies is split into two components as shown in (1). The first part comes from the *apriori* static information, whereas the second part comes by adaptively adjusting the coefficients so that data consistency is maintained.

$$d_{GS}(k) = d_c(k) + \sum_m c_m d_c(k - m.\Delta k) \tag{1}$$

where, d_{GS} is the Generalized series estimate, d_c is the Fourier transform of the static image, c_m are the generalized series coefficients and Δk refers to the spatial-frequency resolution. A fast version of this algorithm outlined in [5] is used for implementation. After this extrapolation, it follows that the deviation of the training data from the ideal, full *k*-space training data decreases. We expect better training data to translate to better training maps in *y*-*f* space.

Phase constraints: The second change proposed is the incorporation of phase constraints from the training map. The training map, though not of best possible resolution, however does contain unaliased signal distributions. Hence, we use the phase information of the training map in estimating the true *y*-*f* map.

$$\Theta = \angle \rho_{train} \tag{2}$$

$$\tilde{\rho} = |\rho| \exp(i\Theta) \tag{3}$$

where, $\tilde{\rho}$ is the final estimate of the signal distribution in *y*-*f* plane. ρ_{train} is the training map.

3 Results and Discussion

3.1 Data Description

fMRI data was obtained for experiments with "visual stimulus" While a subject performed the experiment, 3 two-dimensional T*2-weighted images, each with 64 scans, were acquired using a gradient-echo FLASH sequence (TE/TR 40msec/80.5msec, matrix = 128 × 64; The image matrices were zero-filled to obtain 128 × 128 images with a spatial resolution of 1.953 × 1.953 mm; slice thickness = 5-mm and 2-mm gap). The corresponding two-dimensional anatomical slices were also acquired with a T1-weighted IR RARE sequence (TI = 900 msec; TE/TR 3900msec/40msec, matrix = 512 × 512) in the same experiment session. In all experiments, ON and OFF stimuli were presented at a rate of 5.162 sec/sample. Each stimulation period had four successive stimulation ON states followed by four stimulation OFF states. The stimulations were repeated for eight cycles (total experiment time = 5.5 min), and experiments were carried out at different sessions with different subjects. The visual stimulation task comprised an 8-Hz alternating checkerboard pattern with a central fixation point projected on a LCD system. The subjects were asked to fixate on the point during stimulations. Images were acquired at three axial levels of the brain at the visual cortex.

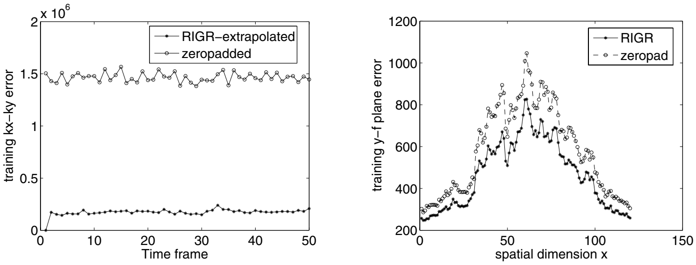
3.2 Performance Evaluation

fMRI images are mainly studied for the activation maps which interpret the information contained in the entire time series of images. Hence, to evaluate the reconstruction performance, we compare the activation maps obtained against the reference activation map. Statistical Parametric Mapping (SPM) is the most widely used method for fMRI time-series analysis [10]. The software package SPM2, that implements SPM, downloaded from [11], was used for analysis. The primary objective is to detect activated voxels and the resulting statistical parametric maps represent the activation strength of each voxel. The scale of the activation-strength obtained is important, since the activation maps are eventually thresholded to obtain truly activated regions. Hence when drastic changes in the scales of activation-strength are observed, the activation maps are considered degraded. Root Mean square error (RMSE), correlation with reference, and mean activation level of the activation maps are used to quantify the degradation in activation. If we analyze the true image time series A and the reconstructed series B, using same SPM method and parameters, we expect comparable scales in activation strength at similar locations, in the resulting statistical parametric maps S_A and S_B. fMRI time-series are first realigned to remove movement effects using least-squares minimization [10] and then smoothed with Full Width at Half Maximum (FWHM) = 4.47mm, 3D Gaussian kernel to decrease spatial noise. Canonical hemodynamic response function (HRF) plus time and dispersion derivatives is used as basis function and changes in BOLD signal associated with the task were assessed on a pixel-by-pixel basis, using the general linear model and the theory of Gaussian fields as implemented in SPM2. This method takes advantage of multivariate regression analysis and corrects for temporal and spatial autocorrelations in the fMRI data. Voxels in the statistical parametric map based on F-contrast below a threshold of $p \leq 0.05$ are identified as activation, which was corrected for multiple comparisons using family-wise-error (FWE).

3.3 Experimental Results

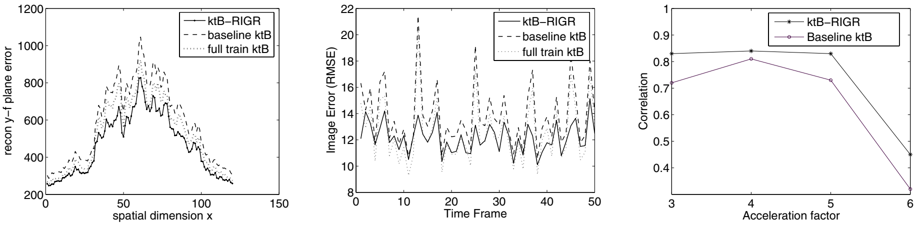
MATLAB was used for all simulations. For our trials, the training and actual acquisitions were generated from the full resolution true k-space, by using the appropriate sub-sampling masks.

In Fig.2 (a), the deviation of the training data with respect to the ideal data is shown in 2 cases. In the first case, the training data is simply zero-padded as



(a) (b)

Fig. 2. Errors for acceleration factor 5 in (a) Training k-space data (b) y-f-Training map



(a) (b) (c)

Fig. 3. Reconstruction errors for acceleration factor 5 in (a) y-f map (b) Image series (RMSE) ; (c) Correlation with reference activation map

in the existing (baseline) k-tB, where as in the second case, the obtained low frequencies are RIGR-extrapolated (proposed). Clearly, the RIGR-extrapolated data is seen to be closer to the truth. In Fig.2 (b), we compare how the gains of Fig.2 (a), translate in the y-f space. It can be observed that the RIGR-extrapolated training map is close to the training map that would have been generated had all the frequencies been available for training (ideal/full training) and is more accurate than the zero-padded map that the original k-tB algorithm uses. In Fig. 3 (a), we see errors in the reconstructed y-f plane as compared to the true y-f plane. The three cases compared are : The training map being ideal (full training), zero-padded (baseline k-tB) and RIGR-extrapolated (proposed). It can be seen that the RIGR-extrapolated case results in lower errors compared to the zero-padded case, consistently for all instants of the time series. In Fig. 3 (b) the time series of errors in RMSE, incurred during image reconstruction in all the three cases outlined above, is shown. It can be seen that the RIGR-extrapolated case and the ideal training map case, are quite comparable, while both consistently outperform the baseline k-tB reconstruction. Fig. 3(c) shows the decline in correlation of the obtained activation map with the reference map, against acceleration factor.

In Fig.4, we observe the activation maps obtained using the two methods, for a gain of factor 5 in temporal resolution. Clearly, the map obtained using Baseline

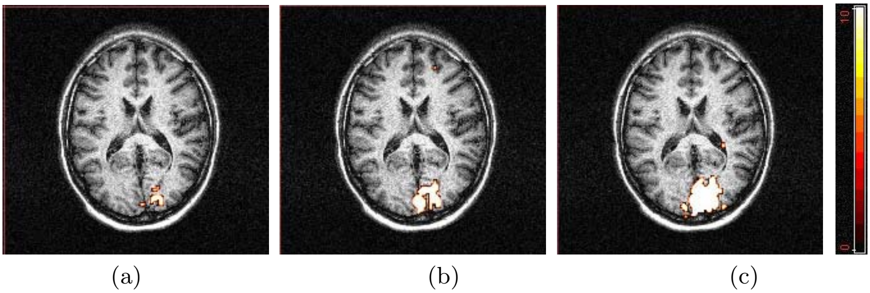


Fig. 4. Thresholded Activation maps obtained using SPM for acceleration factor 5 (a) True Images (b) Proposed method (c) Baseline ktB

k-tB displays more artifacts than the proposed method. We also observe that the gain in PSNR goes upto 10dB. The RMSE of the fMRI time series reduces by about 10% averaged over all time points, with a peak improvement of 35% compared to the existing k-tB for acceleration factors upto 6. For acceleration factor of 6 we notice that the scales of activation maps obtained using baseline k-tB are lower by a factor more than 10, and hence it is not possible to threshold them to see activated regions. On the other hand, the proposed method results in activation maps that are lower by a factor 2 and hence activated regions can be seen at lower thresholds. At accelerations above 6 we notice significant degradation in the strength scales of the activation maps, and hence do not consider them.

We also carried out trials where only one of the two proposed changes were made to the existing algorithm. We first chose to extrapolate training data and skip the incorporation of phase constraints. It was observed that the resulting reconstructions did not show much change when compared against the case where zero-padded training data was used. In this case, we know that the best possible reconstruction achievable is what results out of using the ideal training set. In the next trial, we retained the zero-padded training map, and incorporated only the phase constraint. It was seen that this worsens the performance of the baseline k-tB, since the phase map imposed is a blurred version of the original. Hence, it is observed that incorporating both changes leads better reconstruction compared to the baseline k-tB.

4 Conclusion

In this paper, we have proposed an improved version of the existing dynamic imaging technique k-tB. The changes include improvement in the training map that serves as an estimate to obtain the true signal distribution. The other proposed change is the utilization of the phase-constraints from the training map, rather than the aliased map. Trials on real fMRI data have shown that these 2 changes together lead to improved reconstructions and acceleration factors of upto 6. The reconstruction performance is evaluated using activation maps obtained. We observe upto 10dB improvement in PSNR of activation maps. The proposed technique results in more accurate activation maps and also the image time series incurs mean RMSE of less than 10% averaged over the entire time series, for acceleration factors upto 6.

References

1. Pruessmann, K.P., Weiger, M., Scheidegger, M.B., Boesiger, P.: Sense : Sensitivity encoding for fast mri. Magnetic Resonance in Medicine **42** (1999) 952–962
2. Sodickson, D.K., Manning, W.J.: Simultaneous acquisition of spatial harmonics (smash): ultra-fast imaging with radio frequency coil arrays. Magnetic Resonance in Medicine **38** (1997) 591–603
3. van Vaals, J.J.: Keyhole method for accelerating imaging of contrast uptake. Journal of Magnetic Resonance Imaging **3** (1993) 671–675
4. Liang, Z.P., Lauterbur, P.C.: An efficient method for dynamic magnetic resonance imaging. IEEE Transactions on Medical Imaging **13** (1994) 677–686
5. Liang, Z.P., Madore, B., Glover, G.H., Pelc, N.J.: Fast algorithms for gs-model-based image reconstruction in data-sharing fourier imaging. IEEE Transactions on Medical Imaging **22** (2003) 1026–1030
6. Madore, B., Glover, G.H., Pelc, N.J.: Unaliasing by fourier-encoding the overlaps using the temporal dimension (unfold), applied to cardiac imaging and fmri. Magnetic Resonance in Medicine **42** (1999) 813–828
7. Tsao, J., Boesiger, P., Pruessmann, K.P.: k-t blast and k-t sense: Dynamic mri with high frame rate exploiting spatiotemporal correlations. Magnetic Resonance in Medicine **50** (2003) 1031–1042
8. Tsai, C.M., Nishimura, D.G.: Reduced aliasing artifacts using variable-density k-space sampling trajectories. Magnetic Resonance in Medicine **43** (2000) 452–458
9. Hansen, M.S., Kozerk, S., Pruessmann, K.P., Boesiger, P., Pedersen, E.M., Tsao, J.: On the influence of training data quality in k-t blast reconstruction. Magnetic Resonance in Medicine **52** (2004) 1175–1183
10. Friston, K.J., Holmes, A.P.: Statistical parametric maps in functional imaging: A general linear approach. Human Brain Mapping **2** (1995) 189–210
11. http://www.fil.ion.ucl.ac.uk/spm/software/spm2/.

A Fast Directed Tree Based Neighborhood Clustering Algorithm for Image Segmentation

Jundi Ding[1], SongCan Chen[1], RuNing Ma[2], and Bo Wang[1]

[1] Nanjing University of Aeronautics and Astronautics,
College of Information Science & Technology, 210016, P.R. China
s.chen@nuaa.edu.cn
http://parnec.nuaa.edu.cn
[2] Nanjing University of Aeronautics and Astronautics,
College of Science, 210016, P.R. China

Abstract. First, a modified Neighborhood-Based Clustering (MNBC) algorithm using the directed tree for data clustering is presented. It represents a dataset as some directed trees corresponding to meaningful clusters. Governed by Neighborhood-based Density Factor (NDF), it also can discover clusters of arbitrary shape and different densities like NBC. Moreover, it greatly simplify NBC. However, a failure applying in image segmentation is due to an unsuitable use of Euclidean distance between image pixels. Second, Gray NDF (GNDF) is introduced to make MNBC suitable for image segmentation. The dataset to be segmented is all grays and thus MNBC has the constant computational complexity O(256). The experiments on synthetic datasets and real-world images shows that MNBC outperforms some existing graph-theoretical approaches in terms of computation time as well as segmentation effect.

1 Introduction

Neighborhood-Based Clustering (NBC) algorithm [1] proposed by Zhou S. G. etc is a good data clustering algorithm and can discover clusters of arbitrary shape and different densities using the neighborhood relationship among data points. Experiments in [1] show that NBC is advantageous over DBSCAN [2] in both clustering effectiveness and efficiency. However, in order to develop the algorithm the authors introduced thirteen pre-requisite definitions including the neighborhood based density factor (NDF). Besides, they incorporated the cell-based structure and VA file [3] for clustering very large and high dimensional databases. The two aspects mentioned above make NBC conceptually and structurally complex. In addition, NBC fails in segmenting an image due to an unsuitable use of Euclidean distance between image pixels.

In fact, we just require three key definitions from the thirteen basic ones in NBC, i.e. k-neighborhood, reverse k-neighborhood and NDF, and additionally borrow the idea of directed tree to develop a modified NBC (MNBC) for data clustering, which not only simplifies NBC but also can discover clusters of arbitrary shape and different densities like NBC. It represents a dataset as some directed trees corresponding to meaningful clusters. So, its goal is to find the

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 369–378, 2006.

numbers of directed trees constructed in a top-down strategy. On the other hand, we introduce Grayscale k-neighborhood, Grayscale reverse k-neighborhood and Grayscale NDF (GNDF) and apply MNBC to image segmentation. GNDF characterizes the local density of a gray scale's neighborhood in a relative sense. And MNBC governed by GNDF takes the 256 intensities in a common gray image $I_{m \times n}$ (encoded with 8-bit resolution, m and n are the numbers of rows and columns respectively) as the dataset to be segmented and accomplishes segmentation fast and efficiently. Its computational complexity is O(256), which is independent of the size of the image $m \times n$.

There is some of the related work to our approach: early graph-based methods (EGA) [6], spectral clustering algorithms (SCA) [4], [5], minimum spanning trees (MST) based clustering algorithm [7], [8]. EGA is to generate directed trees for data clustering with a bottom-up process and also guided by a single-scalar control variable but the user must specify it by cross-validation. Its computational complexity is O(N^2). SCA cluster points using eigenvectors of affinity matrices derived from the data set. While powerful, computational cost remains a major obstacle for real-time applications. Its computational complexities is O(N^3). MST based clustering algorithm is a greedy one for segmenting images based on intensity differences between neighboring pixels and requires O($MlogM$),where M is the number of edges in the graph.

The remainder of this paper is organized as follows: Section 2 gives an overview of NBC and refines its three key definitions. The three key definitions avail to design MNBC. Section 3 describes MNBC in detail and presents the evaluation results on some synthetic toy datasets with (EGA) [6], (SCA) [4], [5] and MST [7], [8] to show the good performance of MNBC. Section 4 introduces GNDF and details MNBC for image segmentation, while Sect. 5 delivers comparisons on real world images with MST [7], [8], and Sect. 6 concludes the whole paper.

2 Review of NBC

NBC algorithm [1] uses the neighborhood relationship among data points to build a neighborhood based clustering model with goal to discover clusters of arbitrary shape and different densities. In the description of NBC algorithm, the authors had to introduce thirteen pre-requisite definitions including the neighborhood based density factor (NDF). Here we refine its thirteen basic concepts into just three ones: k-neighborhood, reverse k-neighborhood and NDF. The three key definitions facilitate to design MNBC based on the directed tree. Given a dataset, $X = \{x_1, x_2, \cdots, x_N\}$, N is the size of the d-dimension data set. Euclidean distance between x and y is denoted by dist(x, y).

Definition 1. *(k-Neighborhood) The k-nearest neighbors set of x (kNN(x)) is a set of k nearest neighbors of x($k > 0$), then the x's k-neighborhood (kNB(x)) is the set of objects that lie within the circle region with x as the center and r as the radius, where r is the maximal distance of between x and kNN(x), i.e. $\exists z \in kNN(x)$, $r = dist\ (x,\ z)$, s.t. $\forall y \in kNN(x)$, $dist(x, y) \leq r$.*

Definition 2. *(Reverse k-Neighborhood) The reverse k-neighborhood of x (R-kNB(x)) is the set of objects whose k neighborhood contain x, which can be formally represented as* $R\text{-}kNB(x) = \{y \in X : x \in kNB(y)\}$

Definition 3. *(Neighbor-based Density Factor) The neighbor-based density factor of data point x, denoted by NDF(x),is evaluated as follows:*

$$NDF(x) = \frac{|R\text{-}kNB(x)|}{|kNB(x)|} \tag{1}$$

In practice, $|k\mathrm{NN}(x)|$ is around k for a given single-scalar control variable k. According to Definition 1, it may be a little greater but not less than k. $|\mathrm{R}\text{-}k\mathrm{NB}(x)|$ is quite discrepant for different data points. As a result, there are three situations for NDF(x): larger than 1 (dense point), equal to 1 (even point) and less than 1 (sparse point) [1]. In MNBC, the data points with NDF(x) ≥ 1 are seed nodes, which could be taken as a root node while the others with NDF(x) < 1 can only be taken as leaf nodes or outlier nodes appearing in no directed trees.

3 MNBC

In this section, we describe MNBC. EGA [6] generates directed trees in a bottom-up process and while MNBC adopts a top-down process to construct the directed trees. We will begin by discussing the concepts of graph theory (see [9] and [6]) which are pertinent to MNBC in Sect.3.1 and then proceed to construct the direct trees in Sect.3.2. The evaluation results on some synthetic toy datasets are presented in Sect.3.3.

3.1 The Concepts of Graph Theory

Definition 4. *(Directed Graph and Directed Path) A directed graph is a set of nodes and arcs, each arc leading from an initial node A to a final node A'. A set of arcs* $e_1, e_2, \cdots, e_n$ *is said to be a directed path from A to A', if A is the initial node of* e_1*, A' is the final node of* e_n*, and the final node of* e_k *is the initial node of* e_{k+1} *for* $k = 1, 2, \cdots, n-1$.

Definition 5. *(Directed Tree) A directed tree is a directed graph satisfying 1) Every node A ≠ R is the final node of exactly one arc; 2) L is the initial node of no arc; 3) R is the final node of no arc; 4) There is no directed path from a node A to itself (i.e. no cycles).*

The nodes R and L are called the root and leaf of the directed tree respectively. The final node of the arc whose initial node is A is called the child node of A, denoted C(A). Notice that the root of a directed tree must be unique but the leaf of a directed tree can be more than one, and a path from the root to one of the leaves in a directed tree is unique and consists of the arcs from R to one C(R), C(R) to one C(C(R)), etc.

3.2 Construction of the Directed Trees Based on NDF

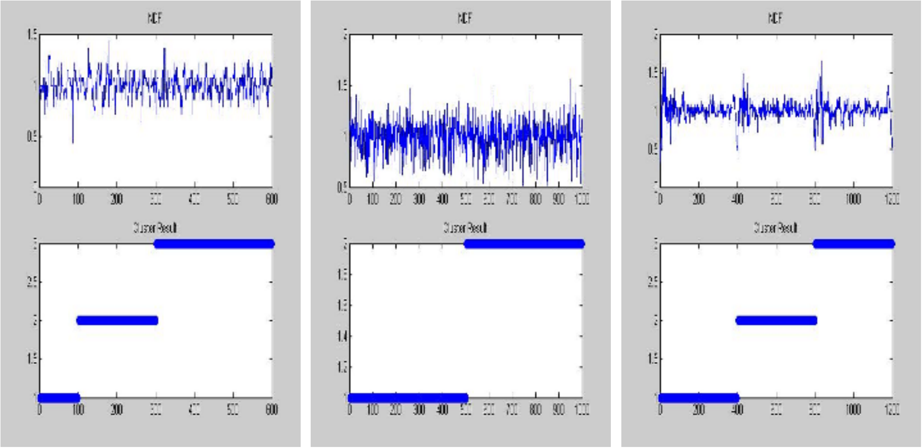
For a given k, MNBC is made up of three phases:
(P1) Computing all kNB(x), R-kNB(x) and NDF (x) according to (1);
(P2) Constructing all directed trees based on NDF evaluated in P1;
(P3) Nodes exist in no directed tree are called outliers.
Obviously, one or more directed trees can be constructed in P2, dependent of the single variable k. The following algorithmic steps summarize P2:

```
1. Initially, $numT = 0$, $V = \{x : NDF(x) \geq 1, x \in X\}$;
2. While $V \neq \emptyset$, an arbitrary $x \in V$ is taken as a root node to construct $T_x$
   (the directed tree of $x$):
   $T_x = \emptyset$; $C(x) = kNB(x)$; $T_x = \{x\} \cup C(x)$;
   $Y = \{y : y \in C(x), NDF(y) \geq 1\}$;
      While $Y \neq \emptyset$,
         For each seed node $y \in Y$, $C(y) = \{z : z \in kNB(y), z \notin T_x\}$
               If $C(y) = \emptyset$, $y$ becomes a leaf node of $T_x$;
               Else $y$ is a root node of the subtree $T_y, T_y = \{y\} \cup C(y)$;
               End
         End
         $C(C(x)) = \bigcup_{y \in Y} C(y), T_x = T_x \cup \bigcup_{y \in Y} T_y$;
         $C(x) = C(C(x))$; $Y = \{y : y \in C(x), NDF(y) \geq 1\}$;
      End
   $numT = numT + 1; X = X \backslash T_x; V = \{x : NDF(x) \geq 1, x \in X\}$;
   End
```

Complexity. The time complexity of P1 is O(N^2) because the most time-consuming work in P1 is the evaluation of kNB queries, which takes O(N^2). The recursive procedure of constructing the directed trees to discover clusters takes O(N) with only three key definitions, i.e. the time complexity of P2 is O(N). Therefore, the total computational complexity of MNBC is O(N^2). MNBC, robust to the order of the initial node selection, has the outstanding capability of discovering all clusters of arbitrary shape and recognizing the outlier points as well as NBC [1].

3.3 Synthetic Datasets and Experimental Results

To evaluate the performance of MNBC for data clustering, we compare it with EGA [6], SCA [4], [5] and MST [7], [8] on three synthetic datasets: three concentric circles (out-circle: 300 points; mediate-circle: 200 points; inner-circle: 100 points), two half circles (each has 500 points) and three spirals (each has 400 points). The NDF values of all data points in the respective dataset and the cluster results of MNBC identified by label are put in Fig.1, which shows that MNBC does not cluster data wrongly. The experimental results are illustrated in Fig.2. For each method, its parameters are tuned over a range in which their clustering results for the three toy data sets are different. To make a fair comparison, we carefully



(a) (b) (c)

Fig. 1. The NDF curves and cluster labels in MNBC for (a) Three concentric circles (N=600, k=14); (b) Two half circles (N=1000, k=50); (c) Three spirals (N=1200, k=25)

choose those parameters for each dataset which make each method work best. From Fig.2, EGA and SCA perform poorly for the three synthetic toy data sets; whereas MNBC and MST have a good structural representation, especially the clustering results by MNBC are identical to the original synthetic data sets as show in the first row of Fig.2. Therefore, MNBC outperforms the others.

4 GNDF and MNBC for Image Segmentation

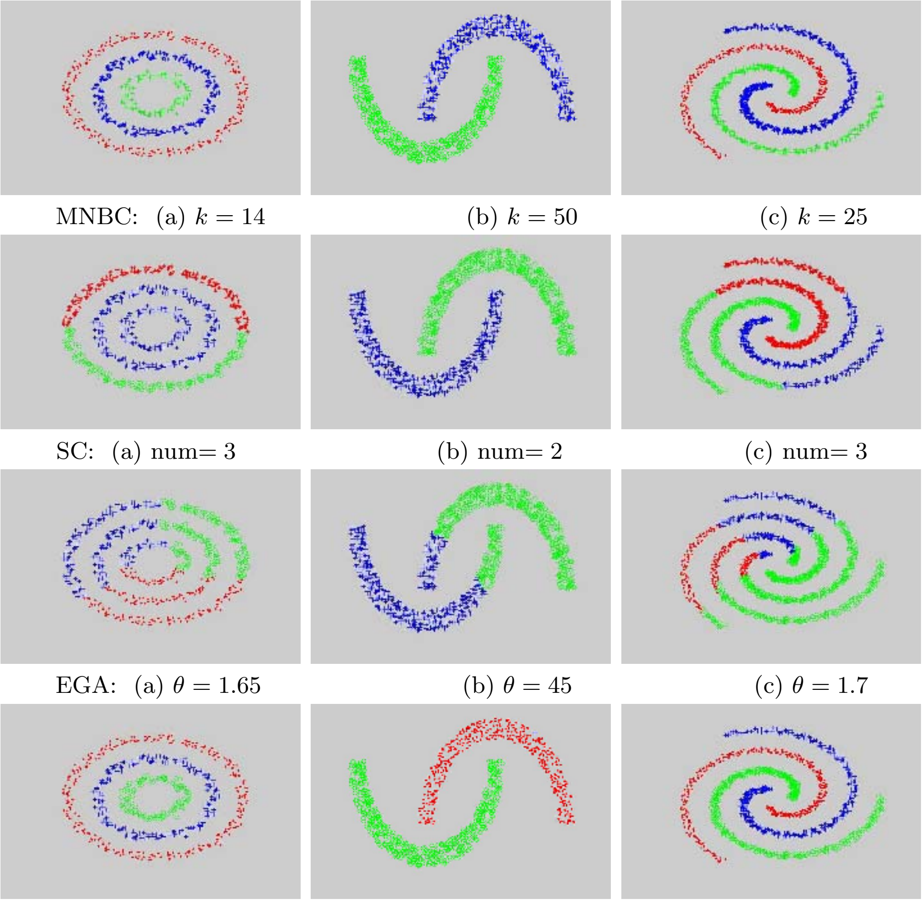
Image segmentation is the most essential and important step of any low-level vision system. In general, a common gray image $I_{m\times n}$ is encoded with 8-bit resolution and has at most 256 grays (m and n are the number of rows and columns respectively). To apply MNBC to segment an image fast and efficiently, we introduce Grayscale k-neighborhood, Grayscale reverse k-neighborhood and Grayscale NDF (GNDF), which characterizes the local density of a gray's neighborhood in a relative sense. MNBC governed by GNDF takes the 256 intensities as the dataset to be segmented and has the computational complexity O(256), which is independent of the size of the image $m \times n$.

4.1 Grayscale Neighborhood-Based Density Factor (GNDF)

Suppose $I = \{0, 1, \cdots, N\}, 0 \leq N \leq 255$, then GNDF is given in Definition 6.

Definition 6. *(Grayscale Neighborhood-based Density Factor)*

$$GNDF(q) = \frac{|R\text{-}kNB(q)|}{|kNB(q)|}, q = 0, 1, \cdots, 255 \tag{2}$$



MNBC: (a) $k = 14$ (b) $k = 50$ (c) $k = 25$

SC: (a) num= 3 (b) num= 2 (c) num= 3

EGA: (a) $\theta = 1.65$ (b) $\theta = 45$ (c) $\theta = 1.7$

MST: (a) $d = 0.4, k = 100$ (b) $d = 10, k = 300$ (c) $d = 1, k = 100$

Fig. 2. Clustering results by MNBC (1st row), SC (2nd row), EGA (3rd row) and MST (4th row), respectively

Denote $num(q)$ as the number of pixels whose intensity is q and $l(q)$ as a natural number satisfying $(num(q) \neq 0)$

$$l(q) = \min \left\{ l(q) \geq 0; \sum_{x \in I, |x-q| \leq l(q)} num(x) \geq k \right\} \tag{3}$$

Then

$$kNB(q) = I \cap \{q - l(q), q - l(q) + 1, \cdots, q + l(q) - 1, q + l(q)\}, \tag{4}$$

$$R\text{-}kNB(q) = \{y \in I; y \in kNB(q)\}, \tag{5}$$

where $kNB(q)$ and $R\text{-}kNB(q)$ are k-neighborhood and reverse k-neighborhood of the grayscale q respectively.

Because the grays of an image are consecutive natural numbers, both k-neighborhood and reverse k-neighborhood of arbitrary grays q are the intersection between the truncation of several consecutive natural numbers and X. Further, it is easy to draw a conclusion in Proposition 1 but its proof is left out due to space of limitation:

Proposition 1. *If* $q_1 \leq q_2$*, then* $q_1 - l(q_1) \leq q_2 - l(q_2), q_1 + l(q_1) \leq q_2 + l(q_2)$.

Proposition 1 indicates that the left and right endpoint values of k-neighborhood of gray q both are monotonically increasing with respect to q, which implies that there is some expanded direction of k-neighborhood of the q. Since MNBC is robust to the initial node selection analyzed above, we can select the minimal gray q^* as the initial node to construct a directed tree.

Like NDF, GNDF of a gray will also probably be larger than 1 or equal to 1 or smaller than 1. The grays with GNDF$(q) \geq 1$ are seed nodes, which could be taken as root nodes while the others with GNDF$(q) < 1$ can only be taken as leaf nodes or outlier nodes not residing on any directed trees. Figure 3 illustrates a simple schematic diagram ($k = 200$), e.g. $num(28) = 50 < 200$, then according to (3) and (4), $l(28) = 2$, kNB$(28) = \{26, 27, 28, 29, 30\}$ and $|k\text{NB}(28)| = 282$ because $\sum_{q=27}^{29} num(q) = 134 < 200$ $\sum_{q=26}^{30} num(q) = 282 > 200$; According to (5), R-$kNB(28) = \{27, 28\}$, $|\text{R-}k\text{NB}(28)| = 104$, then GNDF$(28) = 104/282 < 1$ according to (2).

4.2 MNBC for Image Segmentation with GNDF

Similarly, MNBC based on GNDF is also made up of three phases:

(P1') Input k and compute GNDF (q) according to (2), $q = 1, \cdots, N$;
(P2') Construct the directed trees based on GNDF evaluated in P1';
(P3') Assign pixels to a corresponding directed tree constructed in P2' and the pixels with its gray not in any directed trees are designated as outliers.

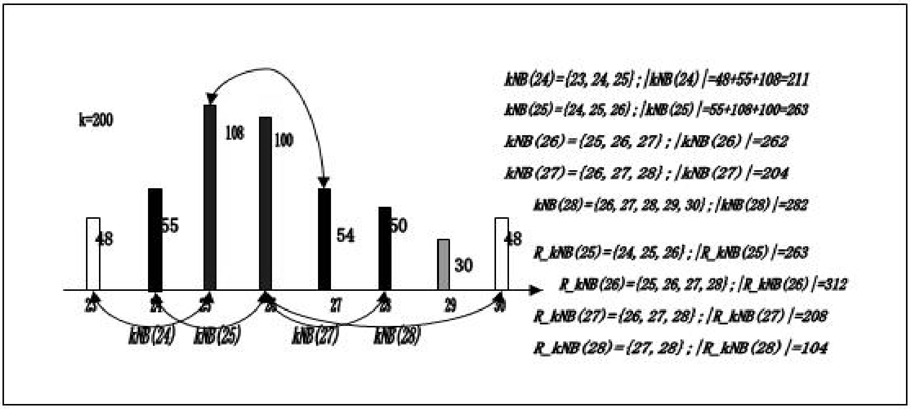

k=200
108
100
48
55
54
50
30
48
23
24
25
26
27
28
29
30
kNB(24)
kNB(25)
kNB(27)
kNB(28)
kNB(24)={23, 24, 25}; |kNB(24)|=48+55+108=211
kNB(25)={24, 25, 26}; |kNB(25)|=55+108+100=263
kNB(26)={25, 26, 27}; |kNB(26)|=262
kNB(27)={26, 27, 28}; |kNB(27)|=204
kNB(28)={26, 27, 28, 29, 30}; |kNB(28)|=282
R_kNB(25)={24, 25, 26}; |R_kNB(25)|=263
R_kNB(26)={25, 26, 27, 28}; |R_kNB(26)|=312
R_kNB(27)={26, 27, 28}; |R_kNB(27)|=208
R_kNB(28)={27, 28}; |R_kNB(28)|=104

Fig. 3. A simple schematic diagram of kNB(q), R-kNB(q)

However, P2' is different from P2. First, the directed trees are constructed with a non-decreasing order, namely, the root of each directed tree T_q is minimal gray q instead of arbitrary q, $q \in I, GNDF(q) \geq 1$. Second, according to Proposition 1, only the maximal seed node $p, p \in C(q), GNDF(p) \geq 1$ is qualified as the root of a subtree of T_q to expand T_q, denoted by T_p, where $C(q)$ is the children nodes of the q, while in P2 all seed nodes must be traversed over x's children nodes to expand T_x. Such a one-direction search makes it for MNBC to segment image easily and fast. P2' is summarized in the following:

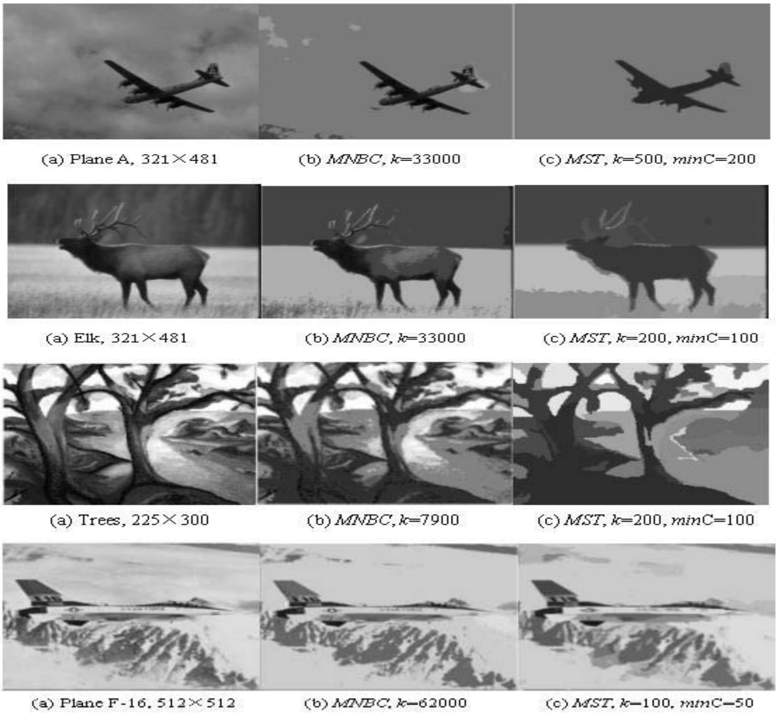
```
1. Initially, $numT = 0$, $V = \{q : GNDF(q) \geq 1, q \in I\}$;
2. While $V \neq \emptyset$, $q^* = minV, q = q^*$, then q is taken as a root node to construct
   $T_q$ (the directed tree of q):
     $T_q = \emptyset$; $C(q) = kNB(q)\backslash q$, $T_q = T_q \cup kNB(q)$;
     $P = \{p : p \in C(q), GNDF(p) \geq 1\}$;
     While $P \neq \emptyset$, $p^* = maxP$; $p = p^*$;
        $C(p) = \{o : o \in kNB(p)\backslash p, o \notin T_q\}$;
        If $C(p) = \emptyset$, p becomes a leaf node of $T_q, T_q = T_q$; break
        Else p is a root node of $T_p$ (a subtree of $T_q$):
              $T_p = kNB(p); T_q = T_q \cup T_p$;
              $C(q) = C(p); P = \{p : p \in C(q), GNDF(p) \geq 1\}$;
        End
     End
   $numT = numT + 1; I = I\backslash T_q; V = \{q : GNDF(q) \geq 1, q \in I\}$;
   End
```

I) Selection of k. The single input parameter k determines the number of regions and the relative size of each region. It can be selected flexibly and purposefully. Let $n_{min} = min_{q \in I} num(q)$, $n_{tot} = \sum_{q \in I} num(q)$. When $k \leq n_{min}$, each gray itself becomes a single cluster. Hence, the number of so-formed regions will be close to 256, which is an over-segmentation problem. In contrast, when $k \geq n_{tot}$, all grays are grouped together to form a single cluster. Thus the number of regions formed is only 1, meaning an under-segmentation. To avoid these two unacceptable extreme cases, we should select k satisfying $n_{min} < k < n_{tot}$. Once k is appropriately selected, the number of regions to be formed is determined automatically.

II) Complexity. The total time complexity of MNBC based on GNDF is O(256), which is independent of the size $m \times n$ of the image $I_{m \times n}$. Because the most time-consuming part of the whole algorithm is the evaluation of kNB queries, which takes only O(256) according to (3) and (4).

4.3 Real Images and Segmented Results

In this subsection, we present three real image experiments to show that MNBC based on GNDF outperforms MST in terms of segmented quality as well as computation time.



(a) Plane A, 321×481 (b) *MNBC*, *k*=33000 (c) *MST*, *k*=500, *min*C=200

(a) Elk, 321×481 (b) *MNBC*, *k*=33000 (c) *MST*, *k*=200, *min*C=100

(a) Trees, 225×300 (b) *MNBC*, *k*=7900 (c) *MST*, *k*=200, *min*C=100

(a) Plane F-16, 512×512 (b) *MNBC*, *k*=62000 (c) *MST*, *k*=100, *min*C=50

Fig. 4. Segmented Results

Figure 4 shows the segmentation results for three real world images, namely, ”Plane A”, ”Elk”, ”Trees” and ”Plane F-16”. Each of them presents different level of difficulties in image segmentation. From left to right, the three columns correspond to respectively the original images, segmented images based on MNBC and MST. The segmentation results are shown with different gray levels representing different segmenting regions. It can be seen that MNBC visually outperforms MST. On one hand, MNBC is capable of preserving details well, such as (1) the letter ”A” in the image ”Plane A”; (2) ”F-16’ mark, the entrance with shape ”□”, the star signature, the text ”US.ATR.FORCE” and ID # ”01568” in the image ”Plane F-16”, whereas MST completely fails. On the other hand, although MST segments the sky correctly as a whole for the image ”Plane A”, the plane A is under-segmented. For the image ”Elk”, MST succeeds in segmenting the body of elk correctly as a whole except for the antler, but the background is over-segmented. For the image ”Trees”, MST has the branches of the trees merged with the riverbank.

5 Conclusion

This paper first presents a MNBC algorithm using the directed tree, which not only can discover clusters of arbitrary shape and different densities like NBC [1] but also simplify NBC greatly. MNBC represents a dataset as some directed trees corresponding to meaningful clusters with just three key definitions refined from NBC. Second, GNDF is defined to make MNBC suitable for image segmentation. Taking all grays in an image as the dataset to be segmented, MNBC has the computational complexity O(256), which is independent of the size of the image. The experiments on synthetic datasets and real-world images shows that MNBC outperforms some existing graph-theoretical approaches in terms of computation time as well as segmentation effect. Our future work will include incorporating the spatial information to MNBC for more effective image segmentation and exploring various applications to which MNBC can be applied.

Acknowledgement

This work is supported by the National Science Foundation of China under Grant No. 60505004, Jiangsu Postdoctoral Science Foundation and the Jiangsu Science Foundation Key Project under Grant No. BK 2004001.

References

1. Zhou, S., Zhao, Y., Guan, J. and Huang, J.: A Neighborhood-Based Clustering Algorithm. PAKDD 2005, LNAI 3518 (1982) 361-371
2. Ester M., Kriegel H., Sander J. and Xu X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In Proc. KDD96 (1996), Portland, Oregon, 226-231 .
3. Weber, R., Schek, H. and Blott,S.: A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In Proc. of VLDB98 (1998) Aug. New York City, NY, 194-205.
4. Shi, J. and Malik, J.: Normalized Cuts and Image Segmentation. Proc. of IEEE Conf. on CVPR (1997) 731-737.
5. Shi, J. and Malik, J.: Normalized Cuts and Image Segmentation. IEEE Trans. on PAMI **22(8)** (2000) 888-905.
6. Koontz, W., Narendra, P. and Fukunaga, K.: A Graph-Theoretic Approach to Nonparametric Cluster Analysis. IEEE Trans. on Comp. **C-25(9)** (1976) Sep. 936-944.
7. Felzenszwalb, P. and Huttenlocher, D.: Image segmentation using local variation. Proc. of IEEE Conf. on CVPR (1998) 98-104.
8. Felzenszwalb, P. and Huttenlocher, D.: Efficient Graph-Based Image Segmentation. International Journal of Computer Vision **59(2)** (2004) Sep.
9. Reinhard, D.: Graph Theory. Electronic Edition (2005) Springer-Verlag Heidelberg, New York.

An Efficient Unsupervised Mixture Model for Image Segmentation

Pan Lin[1], XiaoJian Zheng[1], Gang Yu[2], ZuMao Weng[1], and Sheng Zhen Cai[1]

[1] Faculty of Software of Fujian Normal University,
Fuzhou 350007, P.R. China
linpan99@sohu.com
[2] School of Life Science and Technology, Xi'an Jiaotong University,
Xi'an 710049, P.R. China

Abstract. In this paper, we present an efficient unsupervised mixture model image segmentation method. The idea of this method is that individual image region classes are modeled as mixtures of fuzzy subclasses of mixture distributions, and classification is performed based on the Expectation-Maximization algorithm. To overcome the difficulty of classical mixture model method for noisy image segmentation, spatial contextual information should be taken into account. In particular, the proposed approach based on Markov Random Field was shown to provide more accurate classification of images than traditional Expectation-Maximization algorithm and traditional Markov Random Field image segmentation techniques. The effectiveness of the proposed method is illustrated with synthetic and real images data. The experiments results have shown that the proposed method can achieve more robust segmentation for noisy images.

1 Introduction

Mixture model has widespread applications in image processing and computer vision [1],[2],[3],[4],[5],[6]. Among statistical model image segmentation algorithms, the mixture model has attracted considerable attention in last decade, because for image processing problems, each image region can be characterized by a Gaussian distribution and the entire image can be obtained by describing the image data set with a mixture model. Gaussian finite mixture model is a well-known statistical model for data clustering techniques and image segmentation.

However, the application of finite mixtures model to image segmentation faces some difficulties. First, the estimation of the number of components is still an open question. Second, finite mixture-model based image segmentation technique does not consider image spatial information; this causes the finite mixture model to work only on well-defined images with low levels of noise. In classical mixture statistical model, the each image pixel is associated with exactly one class. This assumption may not be realistic. Some researchers mixed fuzzy and statistical model to solve the problem [7]. These model parameters can be estimated through likelihood maximization using EM algorithm [9]. But the commonly used Maximum likelihood algorithm for image segmentation tends to have an

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 379–386, 2006.

unacceptably large number of misclassified pixels since they ignore spatial contextual information. Markov Random Field(MRF) is considered as a powerful stochastic tool to model the joint probability distribution of the image pixels in terms of local spatial interaction[10], [11],[12],[13],[14].Markov Random Field represents the local characteristics of image structure such that neighboring pixels have a higher probability of being members of the same class. Unsupervised segmentation based on Markov Random Field has been used extensively for the analysis of images segmentation in computer vision. In this work, we present an efficient statistical model to segment an image.

In this paper, we incorporate fuzzy idea into mixture model segmentation scheme. To overcome the difficulty of classical mixture model method for a noisy image segmentation, spatial contextual information should be taken into account. Markov Random Field of prior contextual information is a powerful tool for modeling spatial continuity and other features, and can provide useful information for the image segmentation process. Experiments with synthetic and real images show that the proposed method is more effective for noisy images segmentation problem than traditional method.

The rest of the paper is organized as follows. Section 2, introduces finite mixture model for image segmentation problem. Section 3, describe our proposal to solve image segmentation problem. Section 4, experiments and validate the algorithm.

2 Image Model

The finite mixture model is an efficient clustering analysis tool. Let $X = \{x_1, x_2, \cdots, x_n\}$ be a finite set of pixel of an image. The observed image can be modelled by finite mixture model,The distribution of the image data can be approached by the probability distribution function $p(x_i|\Theta)$ .The mixture model then has the form

$$p(x_i|\Theta) = \sum_{k=1}^{K} \pi_k p_k(x_i|\theta_k) \quad with \quad \sum_{k=1}^{K} \pi_k = 1 \tag{1}$$

where K is the number of image classes,the π_k is mixture weights or mixing coefficient and the parameters of each image class as $\theta_k = (\mu_k, \sigma_k)$ .The set of parameters of a given mixture model is $\Theta = \{\theta_1, \cdots, \theta_k; \pi_1, \cdots, \pi_k\}$.

The density function of the k class region image can be written as

$$p_k(x|\mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp(-\frac{(x-\mu_k)^2}{2\sigma_k^2}) \tag{2}$$

where μ_k and σ_k are the mean and variance of the each image class.The whole image can be described by an independent identically distribution of the X .The likelihood function for an image is

$$p(X|\Theta) = \prod_{i=1}^{N}\sum_{k=1}^{K}\pi_k p_k(x_i|\theta_k) \tag{3}$$

The image segmentation goal will be to estimate vector Θ . Various procedures have been developed for determining the parameters of a mixture of normal densities, often based on the maximum likelihood technique, leading to the EM algorithm. The technique is used to maximize the likelihood function relies on the choice of Θ most likely to give rise to the observed data. In maximum likelihood estimation,the unknown parameter Θ is estimated so that the log-likelihood function as :

$$\hat{\Theta} = \arg\max(\log p(X|\Theta)) \tag{4}$$

The Expectation-Maximization (EM) algorithm[9] is a well-known statistical tool for finding the maximum likelihood estimate(ML) estimate of the mixture model parameters Θ .The EM algorithm used in the analysis consists of the following two steps, namely, the Expectation step and the Maximisation step.

1: The E-Setp(Expectation): compute parts of $Q(\Theta|\Theta^m)$

$$Q(\Theta|\Theta^m) = E[\log p(X|\Theta)|\Theta^m] \tag{5}$$

2: The M-Step(Maximization):search $\Theta^{m+1} = \arg\max Q(\Theta|\Theta^m)$

The above two steps are repeatedly performed until a certain convergence criterion is meet.The iterative EM algorithm for estimating the parameters of the component densities is given by:

$$w_k^m = \frac{\pi_k^m p_k(x_i|\mu_k^m, \sigma_k^m)}{\sum_{k=1}^{K}\pi_k^m p_k(x_i|\mu_k^m, \sigma_k^m)} \qquad \pi_k^{m+1} = \frac{1}{N}\sum_{i=1}^{N} w_k^m \tag{6}$$

$$\mu_k^{m+1} = \frac{\sum_{i=1}^{N} w_k^m x_i}{\sum_{i=1}^{N} w_k^m} \quad (\sigma_k^2)^{m+1} = \frac{\sum_{i=1}^{N} w_k^m |x_i - \mu_k^{m+1}|^2}{\sum_{i=1}^{N} w_k^m} \tag{7}$$

3 The Proposed Segmentation Method

The fuzzy sets,introduced by Zadeh[8]. Let $X = x_1, x_2, \cdots, x_n$ be a set of unlabelled feature vectors x_k .The fuzzy Clustering of data set X into C clusters is characterized by C functions u_{ik} ,the fuzzy partition satisfy the following conditions:

$$u_{ik} : X \to [0,1], i = 1, ..., C \tag{8}$$

and

$$\sum_{i=1}^{c} u_{ik} = 1, i = 1, ..., C \qquad 0 < \sum_{k=1}^{T} u_{ik} < T, i = 1, ..., C \tag{9}$$

These are called membership functions.Because all the components are independent of each other,The whole image can be described by an independent identically distribution of the X . So the corresponding joint pdf is

$$p(X|\Theta) = \prod_{i=1}^{N}\sum_{k=1}^{K} u_{ik}\pi_k p(x_i|\theta_k) \tag{10}$$

where u_{ik} is the fuzzy membership function,from the above equation, we can see that the mixture density is determined by groups,and all the groups are different from each other.

The classical mixture model segmentation method is done each image pixel independently,without taking the classification of its neighbors into account. One common approach is to introduce spatial contextual information for improving segmentation.To modelling the label field image using a Gibbs random field(GRF). Hence the distribution of x is specified by that a Gibbs distribution,

$$p(x) = \frac{1}{Z}\exp\{-\beta\sum_{C} V_C(x)\} \tag{11}$$

where Z is a normalizing constant and the summation is over all cliques C , β is a positive parameter that controls the granularity of the image region. V_C is the potential function. If we consider that a 2-D image is defined on the Cartesian grid and the neighborhood of a pixel is represented by its four nearest pixels then the clique potentials can be defined as

$$V_{ij}(x_i, x_j) = \begin{cases} 1 \ if x_i = x_j \\ 0 \ if x_i \neq x_j \end{cases} \tag{12}$$

This is known as Potts model with an external field V_{ij} ,that weights the relative importance of different class present in the image. The second term takes into account the spatial neighbors information relative to the image data. Here,we define the neighborhood of pixel i ,denote by ∂i ,by 3X3 windows with pixel i being the central pixel. From Eq(10) and Eq(11), the complete-data log likelihood is given by

$$\begin{aligned} L(\Theta) &= \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ik}\log p_k(x_k|\theta_k) + \log p(x) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ik}\log p_k(x_k|\theta_k) - \beta\sum_{C} V_{ij}(x) - \log Z \end{aligned} \tag{13}$$

The EM algorithm for the estimation of the parameters Θ requires that the expectation values u_{ik} of the hidden variables are compute at the E-Step process.

E-step:

$$Q(\Theta, \hat{\Theta}^m) = E[L(\Theta)|u_{ik}^m, \Theta^m] \tag{14}$$

M-Step: Thus ,to computer of the mixture parameters, it can use the same method as it in the M-Step of the EM algorithm.Then the parameters of each image class μ_k^{m+1} and $(\sigma_k^2)^{m+1}$ on the $(m+1)th$ iteration of the EM algorithm are given by

$$\mu_k^{m+1} = \frac{\sum_{i=1}^N u_{ik}^{m+1} x_i}{\sum_{i=1}^N u_{ik}^{m+1}} \quad (\sigma_k^2)^{m+1} = \frac{\sum_{i=1}^N u_{ik}^{m+1} |x_i - \mu_k^{m+1}|^2}{\sum_{i=1}^N u_{ik}^{m+1}} \tag{15}$$

4 Experimental Results

In order to examine the performances more carefully, we use synthetic images ,real images and medical images to compare the experiment performance of the new method present in this paper with the traditional statistical method.

4.1 Noise Synthesis Images Segmentation

The first experiment image is a 256X256 image obtained by adding some gaussian noise to the synthesis image of Fig.1,leading to Fig1.(b).The suggested SNR value is 5.44dB in this example. Fig.1 show 4 class image segmentation. The different segmentation obtained with the different methods are shown in Fig1.We can observe a real visual improvement of results when applying our algorithm. Fig.1(d) shows the EM segmentation results.Fig.1(e) shows classical MRF model segmentation results. By using our proposed method,noisy image can segmented

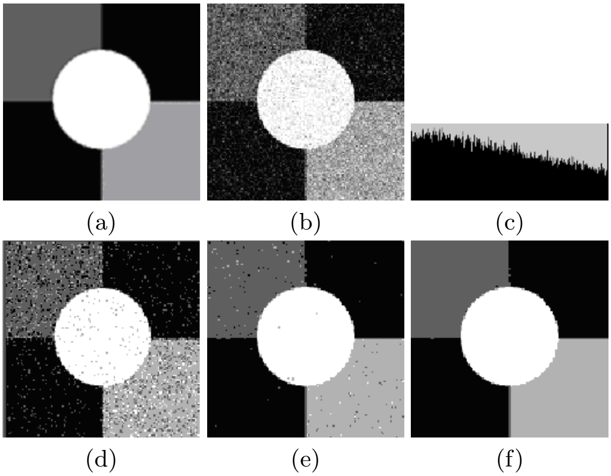


Fig. 1. Synthetic images segmentation.(a)original Image,(b)noise Image,(c)image histogram(d)EM segmentation results,(e)classical MRF model segmentation results,(f)our method results

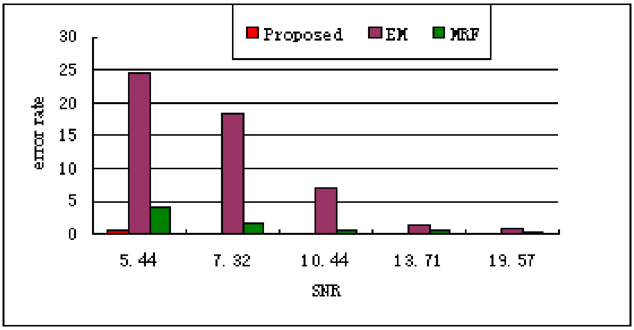
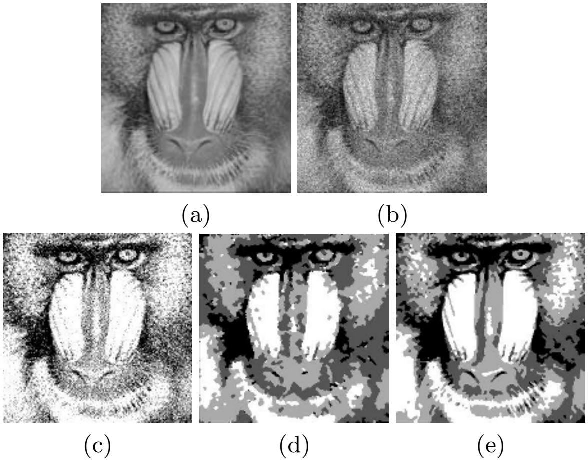


Fig. 2. Performance of the segmentation methods with different levels of noise

well . The result displayed in bottom row of Fig.1(f) demonstrates the parameters of each class are properly estimated and the segmented regions are uniform respectively. This is great improvement over the EM and classical MRF model.We simulated synthesis images with different noise.The quality of segmentation with different levels of noise was analyzed in Fig.2.

4.2 Real Images Segmentation

For this experiment on Baboon image, we consider the 256X256 images of baboon's face presented in Fig.3(a).we add gaussian noise to baboon image shown in Fig.3(b).The value would be suggested in this example. We can observe a real visual improvement of results when applying our algorithm. Clear, this model is enough to capture some finer features of the baoon's face than classical method.


(a) (b)

(c) (d) (e)

Fig. 3. segmentation experiment on an baboon image with 4 class.(c) EM segmentation results,(d) classical MRF model segmentation results,(e) our method results.

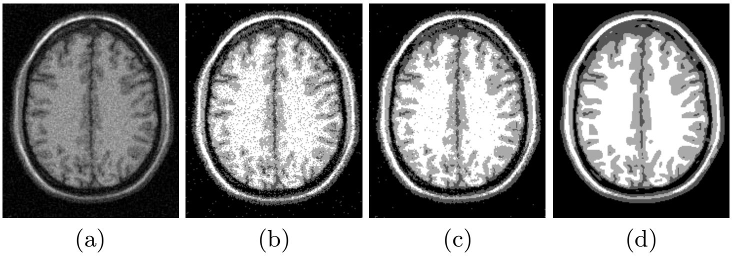


Fig. 4. Performance of the proposed methods for noise MR images.(a)original MR images,(b)EM segmentation results,(c)classical MRF model segmentation results,(d)our method results.

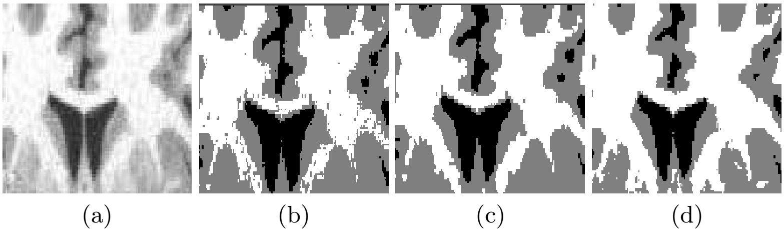


Fig. 5. Performance of the proposed methods for real MR images.(a)original MR images,(b)EM segmentation results,(c)classical MRF model segmentation results,(d)our method results.

4.3 Application in Medical Image

In this experiment, we apply this approach in Magnetic resonance(MR) images.The accurate Segmentation of MR images into different tissue classes, especially gray matter (GM), white matter (WM) and cerebrospinal fluid (CSF), is an important task. for research and clinical study of many neurological pathologies. Fig.5(a)and Fig.4(a) shows two sample slice of real MRI.Fig.5(b) and Fig.4(b) shows the EM segmentation results.Fig.5(c) and Fig.4(c) shows traditional MRF segmentation results,Fig.5(d) and Fig.4(d) shows the segmentation results of our approach are shown above. From these segmentation results, we can see that the segmentation results of our approach are comparable with that of traditional EM and traditional MRF image segmentation method. Our approach has a high ability to resist noise.

5 Conclusions

In this paper, we have presented an efficient unsupervised mixture model image segmentation method, which incorporate fuzzy idea into mixture model. To overcome the difficulty of classical mixture model method for noisy image segmentation,we consider spatial contextual information by incorporating the prior spatial information based on the Markov Random field.We have compared our method with traditional EM and traditional MRF image segmentation techniques.The new algorithm exhibits more reasonable pixel classification and noise suppression performance. We present some examples on synthetic image and real image to illustrate the versatility of our approach. The experimental results show that this method has a significant improvement over classical MRF-based image segmentation. We conclude from the experiments for the synthesis and real images that our algorithm is robust to resist noise.

References

1. G.McLachlan and D.Peel,Finite mixture models, New York:John Wiley,Sons,2000.
2. P.Santago and H.D.Gage, Statistical models of partial volume effect IEEE Trans. Image Process ,1995.4:pp.1531-1540.
3. Penny, W.Bayesian approaches to Gaussian mixture modeling, ,IEEE Trans. Pattern Anal. Mach ,1998.20(11):pp.1133-1142.
4. S. Sanjay-Gopal and T.J.Hebert,Bayesian pixel classification using spatially variant finite mixtures and the generalized EM algorithm,IEEE Trans. Image Process,1998.7(7):pp.1014-1028.
5. M.A.T.Figueiredo,A.K.Jain,Unsupervised learning of finite mixture models, IEEE Trans. Pattern Anal. Mach ,2002.24(3):pp.381-396.
6. K.Blekas,A.Likas,N.P.Galatsanos,A Spatially Constrained Mixture Model for Image Segmentation,IEEE Trans. Neural Networks,2005.16(2):pp.494-498.
7. Gath,I,Geva,A.B,Fuzzy clustering for the estimation of the parameters of the components of mixtures of normal distributions,Pattern Recognition Letters, 1989.9(3): pp.
8. L.A.Zadeh,Fuzzy Sets,Information and Control 8,1965:pp. 338-353.
9. A.P.Dempster,N.M.Laird,D.B.Rubin,Maximum-likelihood from incomplete data via the EM algorithm,J.R.Stat.Soc.Ser.B 39,1977:pp.1-38.
10. S.Z.Li, Markov Random Field Modeling in Computer Vision, New York: Springer-Verlag, 2001.
11. Y.Zhang,M.Brady,and S.Smith,Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm,IEEE Trans. Med. Imag, vol. 20,2001,pp.45-57.
12. S.Geman, D.Geman,Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images,IEEE Trans. Pattern Anal. Mach,1984.6:pp.721-741.
13. J.Besag, Towards Bayesian image analysis, Journal of Applied Statistics,1989.16:pp.395-407.
14. J.Zerubia and R.Chellappa,Mean field annealing using compound Gauss Markov random fields,IEEE Transactions on Neural Networks,1993.4(4):pp.703-709.

Speckle Reduction of Polarimetric SAR Images Based on Neural ICA

Jian Ji[1] and Zheng Tian[1,2,3]

[1] Department of Computer Science & Technology, Northwestern Polytechnical University,
Xi'an, 710072, China
jijiangao@gmail.com
[2] Department of Applied Mathematics, Northwestern Polytechnical University,
Xi'an, 710072, China
[3] Key Laboratory of Education Ministry for Image Processing and Intelligent Control,
Huazhong University of Science & Technology, Wuhan 430074, China

Abstract. The polarimetric synthetic aperture radar (PSAR) images are modeled by a mixture model that results from the product of two independent models, one characterizes the target response and the other characterizes the speckle phenomenon. For the scene interpretation, it is desirable to separate between the target response and the speckle. For this purpose, we proposed a new speckle reduction approach using independent component analysis (ICA) based on statistical formulation of PSAR image. In addition, we apply four ICA algorithms on real PSAR images and compare their performances. The comparison reveals characteristic differences between the studied neural ICA algorithms, complementing the results obtained earlier.

1 Introduction

Recent advances in the remote sensing polarimetric synthetic aperture radar (PSAR) systems provide a rich set of data for the same scene. This set of data brings knowledge on the nature of targets [1]. However, the PSAR images are corrupted by speckle that appears as a granular signal-dependent noise. It has the characteristics of a non-Gaussian multiplicative noise [2]. Due to its granular appearance in an image, speckle noise makes it very difficult to visually and automatically interpret SAR data. Therefore, speckle filtering is a critical preprocessing step for many SAR image processing tasks, such as segmentation and classification [3].

Independent component analysis (ICA) is an unsupervised technique that tries to represent the data in terms of statistically independent variables [4]. ICA has lately drawn a lot of attention both in unsupervised neural learning and statistical signal processing. ICA is suitable for neural network implementation and different theories recently proposed for that purpose lead to the same iterative learning algorithm. Different neural-based blind source separation algorithms are reviewed in [5-9]. The potential application of ICA in remote sensing has been validated, especially in SAR image processing. It can improve the image quality and enhance the performance of pixel classification. In short, ICA algorithm will be a useful method for remote sensing research [10].

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 387 – 393, 2006.

For the same scenario, polarimetric SAR can provide a group of different polarimetric image data, and the characters of target are separated in the images polluted by speckle and are independent to the speckle noise. Thus ICA can be applied to this model and a new method is put forward to reduce speckle. In addition, it is important to know the computational properties of available algorithms in remote sensing applications. This calls for an experimental comparison of the ICA algorithms. In a companion paper [11], it had presented a first comparison of neural ICA algorithms using artificially generated data related blind source separation (BSS) problem. In this paper, we complementing the results obtained earlier by apply the four ICA algorithms to reduce speckle and compare their performance.

2 Model and Statistics of PSAR Image

Let x_i be the content of the pixel in the ith SAR image, s_i the noise-free signal response of the target, and n_i the speckle. Then, we have the following multiplicative model [2]:

$$x_i = s_i \cdot n_i \tag{1}$$

By supposing that the speckle has unity mean, standard deviation of σ_i, and is statistically independent from the observed signal x_i, the multiplicative model in (1) can be rewritten as:

$$x_i = s_i + s_i \cdot (n_i - 1) \tag{2}$$

The term $s_i \cdot (n_i - 1)$ represents the zero mean signal-dependent noise and characterizes the speckle noise variation. Thus, we have converted the multiplicative model into the additive model. The speckle filtering can be considered as the estimation of the unobservable image s_i from the noisy observation x_i.

3 ICA Formulation

The concept of ICA was first proposed by Common [4] in 1994, which has undergone a rapid development. By transforming the input signals, ICA algorithms make the mutual dependency among different signal components minimum. When the mutual dependency among signal components is measured by the different criteria, the different ICA algorithms can be derived.

Let us assume that an array of sensors provides a vector of m observed signals $\mathbf{x} = [x_1, x_2, \cdots, x_m]^T$ that are linear mixtures of $n \le m$ unobserved random processes $\mathbf{s} = [\mathrm{s}_1, \mathrm{s}_2, \cdots, \mathrm{s}_n]^T$ sources. The ICA problem is typically formulated as follows [4].

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{e} \tag{3}$$

where $\mathbf{A}$ is an unknown $m \times n$ full-column rank matrix that represents the mixing system, and $\mathbf{e} = [e_1, e_2, \cdots, e_m]^T$ is the vector of noise components which are assumed in this paper to be Gaussian and statistically independent of the sources.

In order to recover the sources, the observations are processed by a $n \times m$ separating matrix $\mathbf{B}$ to produce the vector of outputs or sources estimation

$$\mathbf{u} = \mathbf{B}\mathbf{x} \tag{4}$$

When the separation is obtained the overall mixing and separating transfer matrix $\mathbf{G} = \mathbf{B}\mathbf{A}$ contains a single nonzero element per row and per column.

In several ICA algorithms, the data vectors $\mathbf{x}$ are preprocessed by whitening (sphering) them: $\mathbf{v} = \mathbf{V}\mathbf{x}$. Here $\mathbf{v}$ denotes the whitened vector satisfying $E[\mathbf{v}\mathbf{v}^T] = \mathbf{I}$, where $\mathbf{I}$ is the unit matrix, and $\mathbf{V}$ is a $m \times n$ whitening matrix. After prewhitening the subsequent $m \times m$ separating matrix $\mathbf{W}$ can be taken orthogonal, which often improves the convergence. Thus in whitening approaches the total separating matrix is $\mathbf{B} = \mathbf{W}\mathbf{V}$.

4 Neural ICA Algorithms

In this paper we concentrate on reduction speckle for PASR image, describing the algorithms included in our study only briefly. For more details, see the references [5-9].

4.1 Natural Gradient Algorithm (NG)

Originally proposed on heuristic grounds [5], this popular and simple neural gradient algorithm was later on derived from information-theoretic criteria [6]. The update rule for the separating matrix $\mathbf{B}$ is

$$\Delta\mathbf{B} = \mu_k[\mathbf{I} - \mathbf{g}(\mathbf{u})\mathbf{u}^T]\mathbf{B} \tag{5}$$

The notation $\mathbf{g}(\mathbf{u})$ means that the nonlinearity $g(t)$ is applied to each component of the vector $\mathbf{u} = \mathbf{B}\mathbf{x}$. The learning parameter μ_k is usually a small constant. The basic algorithm (5) does not use prewhitening, which leads in many cases to a poor convergence. Therefore whitening is often applied to improve the convergence properties.

4.2 Equivariant (EICA) Algorithm

This algorithm is a quasi-Newton iteration that will converge to a saddle point with locally isotropic convergence, regardless of the distributions of sources. It has the following equivariant and robust in respect to Gaussian noise algorithm [7]:

$$\Delta\mathbf{B}(l) = \mathbf{B}(l+1) - \mathbf{B}(l) = \eta_l[\mathbf{I} - \mathbf{C}_{1,q}(y,y)\mathbf{S}_{q+1}(y)]\mathbf{B}(l) \tag{6}$$

where $\mathbf{S}_{q+1}(y) = \mathrm{sign}(\mathrm{diag}(\mathbf{C}_{1,q}(y,y)))$ an $\mathbf{C}_{p,q}(y,y)$ denotes the cross-cumulant matrix whose elements are $[\mathbf{C}_{p,q}(y,y)]_{ij} = Cum(\underbrace{y_i \ \cdots \ y_i}_{p}, \underbrace{y_j \ \cdots \ y_j}_{q})$.

4.3 Extended Information Maximization (Infomax) Algorithm

The purpose of extended information maximization algorithm [8] is, to provide a learning rule with a fixed nonlinearity that can separate sources with sub- and super-Gaussian p.d.f.'s. Employing a strictly symmetric bimodal univariate distribution, obtained by a weighted sum of two Gaussian distributions, given as,

$$p(\mathbf{u}) = \frac{1}{2}(N(\mu,\sigma^2) + N(-\mu,\sigma^2)) \tag{7}$$

leads to the learning rule [4] for strictly sub-Gaussian sources,

$$\Delta\mathbf{W} \infty [\mathbf{I} + \tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T]\mathbf{W} \tag{8}$$

For unimodal super-Gaussian sources, the following density model is adopted,

$$p(\mathbf{u}) \infty N(0,1)\mathrm{sech}^2(\mathbf{u}) \tag{9}$$

which leads to the following learning rule for strictly super-Gaussian sources,

$$\Delta\mathbf{W} \infty [\mathbf{I} - \tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T]\mathbf{W} \tag{10}$$

Therefore, using these two equations, we can obtain a generalized learning rule, using the switching criterion in order to distinguish between the sub- and super-Gaussian sources by the sign before the hyperbolic tangent function as,

$$\Delta\mathbf{W} \infty [\mathbf{I} - \mathbf{K}\tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T]\mathbf{W} \tag{11}$$

where $\mathbf{K}$ is an N-dimensional diagonal matrix composed of k_i's, defined as,

$$k_i = \mathrm{sign}(kurt(u_i)) \tag{12}$$

4.4 Fast Fixed-Point (FastICA) Algorithms

One iteration of the generalised fixed-point algorithm for finding a row vector $\mathbf{w}_i^T$ of $\mathbf{W}$ is [9]:

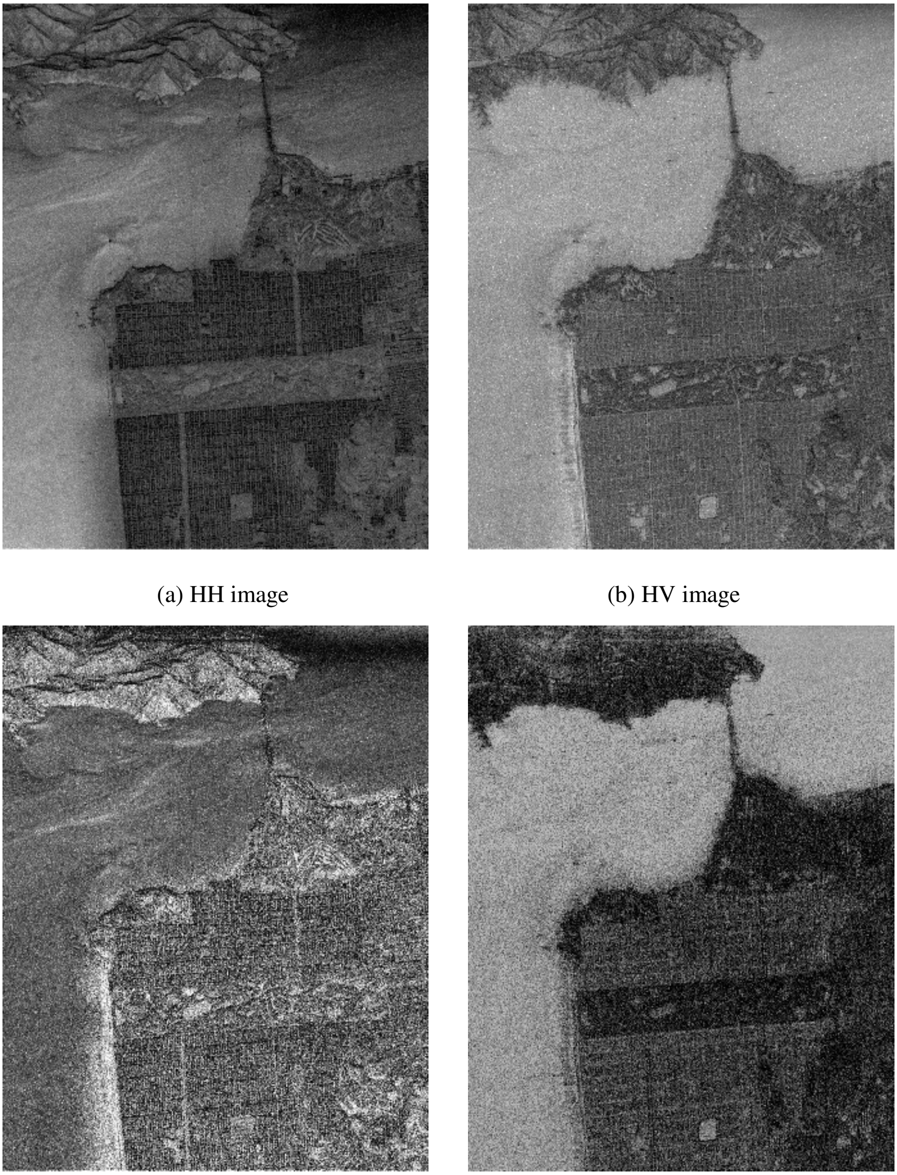
$$\mathbf{w}_i^* = \mathbf{E}\{\mathbf{v}g(\mathbf{w}_i^T\mathbf{v})\} - \mathbf{E}\{g'(\mathbf{w}_i^T\mathbf{v})\}\mathbf{w}_i \tag{13}$$

$$\mathbf{w}_i = \mathbf{w}_i^* / \| \mathbf{w}_i^* \| \tag{14}$$

Here $g(t)$ is a suitable nonlinearity, typically $g(t) = t^3$ or $g(t) = \tanh(t)$, and $g'(t)$ is its derivative. The expectations are in practice replaced by their sample means.Hence the fixed-point algorithm is not a truly neural adaptive algorithm. The algorithm requires prewhitening of the data. The vectors $\mathbf{w}_i$ must be orthogonalised against each other; this can be done either sequentially or symmetrically. Usually the algorithm (13) converges after 5-20 iterations.

5 Simulations and Results

JPL AIRSAR L-band data from San Francisco are used for illustration. This San Francisco scene contains a rich variety of scatterers: specular scattering from the ocean at the top of the scene, double bounce scattering from the city blocks, volume scattering from trees, and surface scattering from grass. The original images are shown in Fig.1, including three polarimetric modes, they are HH (Horizontal- Horizontal),



(a) HH image

(b) HV image

(c) VV image

(d) HH/VV radio image

Fig. 1. L band PSAR images

HV (Horizontal-Vertical) and VV (Vertical-Vertical). In polarimetric datas, the amplitude ratio has a number of important uses, both as a means of inferring physical propertied of a medium and as a way of removing terrain effects. In order to get better speckle reduction image, we also add HH/VV radio image to be the input data.

During the ICA application, pre-processing should be taken: Every image of 700×900 pixels should be transformed to vector, and a 4×630000 matrix was produced from the four images; The data matrix should be normalized in order to transform the pixel intensity from the nature data field to traditional data field.

In order to analyze the ability of speckle reduction by quantity, we define equivalent number of look (ENL), a good approach of estimating the speckle noise level in a SAR image, to measure the performance of speckle intensity over a uniform image region [1]. That is:

$$\mathrm{ENL} = \frac{(\mathrm{mean})^2}{\mathrm{variance}} \tag{14}$$

The ENL is equivalent to the number of independent intensity values averaged per pixel. The larger the ENL, the less the speckle effect and the stronger the ability of speckle reduction.

The output matrix **u** of ICA has become 4×630000. Comparing the four independent components (ICs), the IC 4 is complex noise. Therefore we only compute ENL of IC 1, IC 2 and IC 3. The ability of speckle reduction with different algorithms is showed in Table. 1, the ENL of original images, PCA, NG, EICA Infomax and FastICA were listed. Table 2 shows the runtime of different ICA methods. Compared with the three original images, the ENL of three PCs were increased obviously. But the ENL of PCs is lower than that of ICs. The results shown in Table 1 indicate that the FastICA and EICA algorithms performed best, with NG and Infomax having close values, while PCA is the most remote. In addition, Table 2 shows that the FastICA's speed is fastest, while Infomax has slowest speed.

Table 1. Comparison ENL of different ICA algorithms

Origin	HH mode	4.89		IC 1	24.82
PSAR	VV mode	3.77	EICA	IC 2	17.87
image	HV mode	7.79		IC 3	7.15
	PC 1	16.11		IC 1	29.41
PCA	PC 2	7.71	Infomax	IC 2	9.13
	PC 3	4.58		IC 3	6.61
	IC 1	20.98		IC 1	28.76
NG	IC 2	10.49	FastICA	IC 2	16.24
	IC 3	9.79		IC 3	6.38

Table 2. Comparison runtime of different ICA algorithms

Algorithm	NG	EICA	Infomax	FastICA
Runtime(s)	687	92	1267	23

In conclusion, the original images were improved after PCA processing, and four ICA are efficient optimizing algorithm. After ICA processing, IC 1 is the best component, the speckle index is decreased more, and the speckle is farthest separated from the original images. Comparing with other algorithms, FastICA is a fast and efficient method.

6 Conclusions

Based on rigorous statistical formulation of PSAR image, a new speckle reduction approach using ICA is proposed. In addition, we apply four ICA algorithms on real PSAR images and compare their performances. The experiment shows that ICA has effectively reduced the speckle noise of SAR image, has improved the image quality and manifested its strong ability in image separation. ICA has been widely used in blind source separation, but it is not widely used in image processing and is rarely used in remote sensing. We expect that ICA will be widely applied in remote sensing and will accelerate the development of it.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (60375003) , Aeronautics and Astronautics Basal Science Foundation of China (03I53059).

References

[1] Oliver, C. and Quegan, S.: Understanding Synthetic Aperture Radar Images. Artech-House, London, 1998.

[2] Chitroub, S. Houacine, A. and Sansal, B.: Statistical characterisation and modelling of SAR images. Signal Processing, Vol. 82, No. 1, (2002) 69-92.

[3] Pi, Y. et al.: Polarimetric speckle reduction using multi-texture maximum likelihood method. IEE Electronic Letter, 39(2003) 18, 1348-1349.

[4] Common, P.: Independent component analysis, a new concept? Signal processing. 1994,36:287-314.

[5] Cichocki, A. and Unbehauen, R.: Robust neural networks with on-line learning for blind identification and blind separation of sources. IEEE Trans. on Circuits and Systems, 43(11): (1996) 894-906.

[6] Yang, H. and Amari, S. -I.: Adaptive online learning algorithms for blind separation: Maximum entropy and minimum mutual information. Neural Computation. 9(7): 1457-1482, October 1997.

[7] Cruces, S. Castedo, L. Cichocki. A.: Robust blind source separation algorithms using cumulants. Neurocomputing. vol. 49, (2002) 87-118.

[8] Lee T-W, Girolami M, Sejnowski T J.: Independent Component Analysis Using an Extended Infomax Algorithm for Mixed Subgaussian and Supergaussian Sources [J]. Neural Computation, 1999, 11 (2): 417-441.

[9] Hyvärinen, A.: Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. IEEE Transactions on Neural Networks 10(3): 626-634, 1999.

[10] Fiori, S.: Overview of independent component analysis technique with an application to Synthetic Aperture Radar (SAR) imagery processing, Neural Networks, 16(2003) Special Issue, 453-467.

[11] Giannakopoulos, X. Karhunen, J. and Oja, E.: An experimental comparison of neural ICA algorithms. In Proc. Int. Conf. on Artificial Neural Networks, Skovde, Sweden, 1998.M. Girolami and C. Fyfe. Generalised independent

Robust ICA Neural Network and Application on Synthetic Aperture Radar (SAR) Image Analysis

Jian Ji[1] and Zheng Tian[2,3]

[1] Department of Computer Science & Technology, Northwestern Polytechnical University,
Xi'an, 710072, China
jijiangao@126.com

[2] Department of Applied Mathematics, Northwestern Polytechnical University,
Xi'an, 710072, China

[3] Key Laboratory of Education Ministry for Image Processing and Intelligent Control,
Huazhong University of Science & Technology, Wuhan 430074, China

Abstract. Independent component analysis (ICA) has shown success in the separation of sources in lots of applications. However, in synthenic aperture radar (SAR) images the noise is multiplicative, so the applicability of ICA is seriously reduced. This paper proposes a new robust independent component analysis neural network (RICANN) that improves the robustness of ICA by adding outlier rejection rule. Its application in synthetic aperture radar (SAR) is discussed. The results show the potential usage in SAR image processing problems.

1 Introduction

Synthenic aperture radar (SAR) can penetrate clouds and operate day and night, and image with high resolution, therefore it has important use in the construction of national economy and national defence. But for the speckle consisted in SAR image, good result of SAR image analysis can not be gotten with traditional analysis methods, therefore it is very necessary to study new methods to analysis SAR with speckle and it have wide application foreground.

Independent Component Analysis (ICA) is an unsupervised technique which tries to represent the data in terms of statistically independent variables. ICA has lately drawn a lot of attention both in unsupervised neural learning and statistical signal processing. Most of these methods were developed in the case of noiseless data, and differ from one another in the way they enforce independence. The Maximum Likelihood (ML) method [1] directly assumes a factorized form for the joint source distribution; in the infomax method [2], entropy is used as a measure of independence; other methods ensure independence by minimizing contrast functions related to statistics of order greater than two [3]. The strict relationships among the various methods have been investigated as well [4], and some fast and efficient algorithms have been proposed, such as cumulant ICA [5] and FastICA [6]. Although some of the proposed algorithms have been experimentally shown to perform well even in the lack of

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 394 – 403, 2006.

independence, all of them perform poorly when noise affects the data. Recently, some work has been done to overcome this limitation. In particular, the noisy FastICA algorithm [7], and an Independent Factor Analysis (IFA) method [8] [9] have been developed, the latter being also capable of estimating the noise covariance matrix. Nevertheless, while providing satisfactory estimates of the mixing matrices, these methods still produce noisy source estimates [10].

Here, we propose a robust ICA neural network (RICANN) to SAR image analysis. The objective of this paper is to develop novel algorithms that are more robust with respect to noise than existing techniques or that can reduce the noise in the estimated output vector. After a pre-processing stage means of PCA, we remove outliers by applying outlier rejection rule for multivariate data. Then we apply the ICA method on the clean data set. Finally, we provide experimental results for this algorithm to SAR image separation, and compare its performance with the conventional ICA. We also give an application experiment of multi-frequency polarimetric SAR images enhancement and feature extraction. The results claim that our RICANN method enables us to increase robustness against speckle noise and be effective in SAR image analysis.

2 Classical ICA

Let us assume that an array of sensors provides a vector of m observed signals $x(t)=\left[x_1(t),x_2(t),\cdots,x_n(t)\right]^T$ that are linear mixtures of $n\geq m$ unobserved random processes $s(t)=\left[\mathrm{s}_1(t),\mathrm{s}_2(t),\cdots,\mathrm{s}_m(t)\right]^T$ sources. The problem of ICA is defined for the noise case, where the sources and observations have the following linear relation.

$$x(t)=As(t)+n(t)=\sum\nolimits_{i=1}^{m}a_i s_i(t)+n(t) \tag{1}$$

$A=[a_1,\cdots,a_m]$ is a constant full-rank $n\times m$ mixing matrix whose elements are the unknown coefficients of the mixtures. The vectors a_i are basis vectors of ICA.

In standard neural and adaptive source separation approaches, an $m\times n$ separating matrix $W(t)$ is updated so that the m -vector

$$y(t)=W(t)x(t) \tag{2}$$

becomes an estimate $y(t)=\hat{s}(t)$ of the original independent source signals. Fig. 1 shows a schematic diagram of the mixing and ICA system. In neural realizations, $y(t)$ is the output vector of the network and the matrix $W(t)$ is the total weight matrix between the input and output layers.

With a neural realization in mind, it is desirable to choose the learning algorithms so that they are as simple as possible but yet provide sufficient performance. In this paper, we use the neural rule proposed by S. Cruces [5]:

$$\Delta W(l)=W(l+1)-W(l)=\eta_l[\mathbf{I}-\mathbf{C}_{1,q}(y,y)\mathbf{S}_{q+1}(y)]W(l) \tag{3}$$

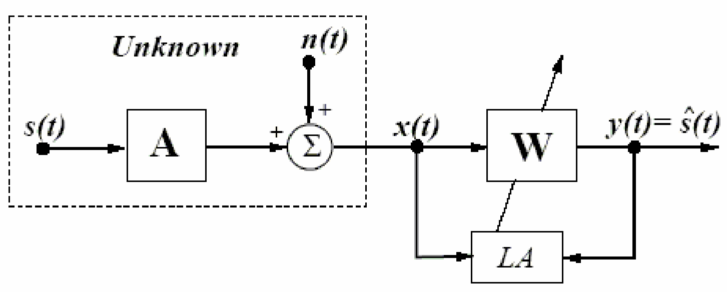


Fig. 1. The mixing model and neural network for ICA. LA means learning algorithm.

where $\mathbf{S}_{q+1}(y) = \mathrm{sign}(\mathrm{diag}(\mathbf{C}_{1,q}(y,y)))$ and $\mathbf{C}_{p,q}(y,y)$ denotes the cross-cumulant matrix whose elements are

$$[\mathbf{C}_{p,q}(y,y)]_{ij} = Cum(\underbrace{y_i \ \cdots \ y_i}_{p}, \underbrace{y_j \ \cdots \ y_j}_{q}) \tag{4}$$

This algorithm is a quasi-Newton iteration that will converge to a saddle point with locally isotropic convergence, regardless of the distributions of sources.

The data vectors $x(t)$ usually are pre-processed using a whitening transformation

$$v(t) = V(t)x(t) \tag{5}$$

here $v(t)$ denotes the whitened vector, and $V(t)$ is a $m \times n$ whitening matrix. In whitening, the matrix $V(t)$ is chosen so that the covariance matrix $E\{v(t)v(t)^T\}$ becomes the unit matrix I_m . Thus the components of the whitened vectors $v(t)$ are mutually uncorrelated and they have unit variance. Uncorrelatedness is a necessary condition for the stronger independence condition. After pre-whitening the separation task usually becomes easier, because the subsequent separating matrix $\widehat{W}$ can be constrained to be orthogonal [4]:

$$\widehat{W}\widehat{W}^T = I_m \tag{6}$$

where I_m is the $m \times m$ unit matrix.

3 The Robust ICA Neural Network (RICANN)

Because the result of ICA can be affected a lot by outliers in the data, we want to avoid this sensitivity and remove the worst outliers in a preprocessing step. We will try two rules for flagging outliers in the raw data [11]. They are based on different distances or outlying measures computed at each data point. The corresponding rejection rule then flags all points whose outlyingness exceeds a certain cutoff value.

First the data points of the data matrix x_i are projected on a subspace defined by means of a measure of outlyingness. This measure is obtained by projecting the data points on many univariate directions z. For every direction a robust center and scale of the projected data points $x_i'z$ is computed, namely the univariate Minimum Covariance Determinant (MCD) estimator [12] of location $\hat{\mu}^i_{\text{MCD}}$. and scale $\hat{\sigma}^i_{\text{MCD}}$. The outlyingness of a data point x_i is then measured by means of:

$$\text{outl}(x_i) = \max_{z \in B} \frac{| x_i' z - \hat{\mu}^i_{\text{MCD}} |}{\hat{\sigma}^i_{\text{MCD}}} \tag{7}$$

where B contains all directions(unit length vectors) we search over. Then we obtain a subspace with smallest outlyinhness that fits the data well. We project the data points on this subspace where we robustly estimate their location and their scatter matrix by means of the MCD estimator, of which we compute its m non-zero eigenvalues $l_1, \cdots, l_m$. The corresponding eigenvectors are the m robust principal components. Formally, writing the (column) robust eigenvectors next to each other yields the $n \times m$ matrix P with orthogonal columns. The location estimate is denoted as the column vector $\hat{\mu}$ and called the robust center. Thus, projecting the observations onto this subspace yields the scores t_i satisfy

$$t_i = (x_i - \hat{\mu}')P \tag{8}$$

To distinguish between regular observations and the outliers, we take into account the orthogonal distance OD_i of each observation:

$$\text{OD}_i = \| x_i - \hat{\mu} - P t_i' \| \tag{9}$$

The first rejection rule flags all points whose robust distance OD_i exceeds a cutoff value.

We also consider the score distance SD_i which represents the distance inside the PCA space taking into account the covariance structure of the data. More formally this distance is defined by:

$$\text{SD}_i = \sqrt{t_i^T L^{-1} t_i} \tag{10}$$

where L is the diagonal matrix with the eigenvalues $l_1, \cdots, l_m$. The corresponding rejection rule flags all points whose outlyingness SD_i exceeds a cutoff.

Fig. 2 shows a three-layer neural network for RICANN, where the first layer performs pre–whitening (sphering), the second layer is flag noisy using rejection rule and the third one - separation of sources. The operation of the network is described by

$$y(t) = \widehat{W}(t)v(t) = \widehat{W}Vx(t) = W(t)x(t) \tag{11}$$

where $W = \widehat{W}V$ is the total separating matrix.

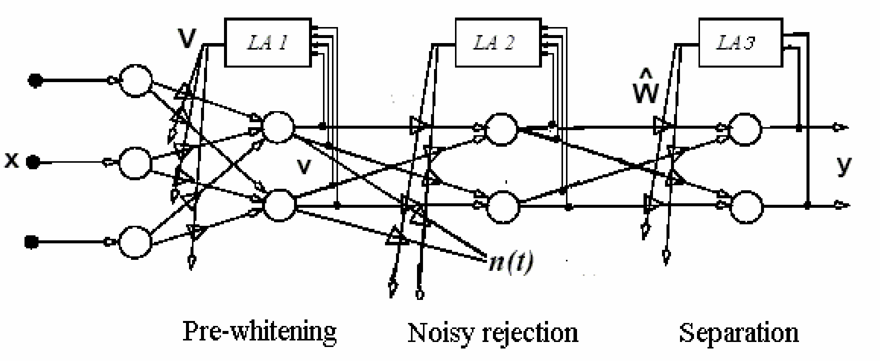


Fig. 2 The three–layer robust network for pre–whitening, noisy rejection and blind separation

4 Experiments with Blind Separation SAR Images

Most of the geological and vegetative ground surfaces are not homogeneous. The pixels resulting from the scanned images of the mentioned zones are, therefore, formed of a mixture of spectral signatures. In theory, one estimates that the global radiometric value of pixels is equal to the contribution average of electromagnetic radiation, emitted or reflected by the study surfaces. Thus, the needed information is not immediately provided by the single radiometric pixel value. For this reason, the mixed pixels are often sources of uncertainty and inaccuracy [13]. This is a situation that seems suited for handling by blind source separation (BSS) techniques. In this section, the performance of the RICANN algorithm is demonstrated using two separation examples(case Ⅰ and case Ⅱ).

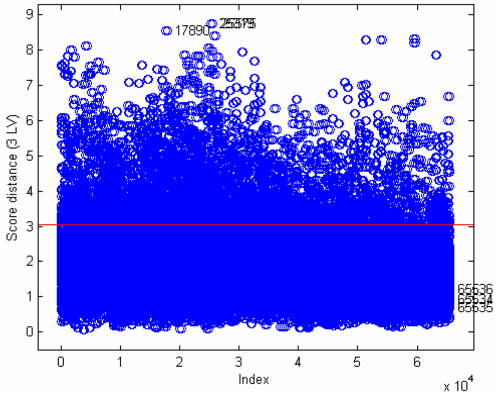
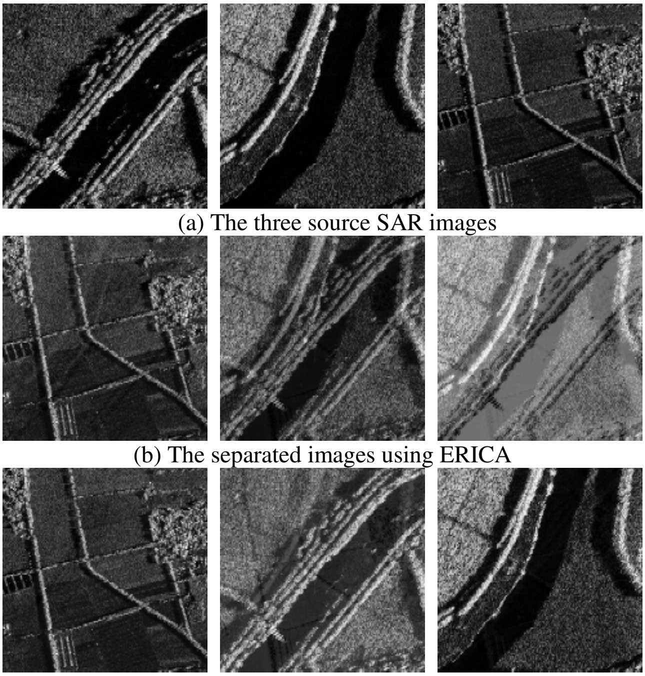


Fig. 3. The score diagnostic plot of example Ⅰ

4.1 Case Ⅰ

Here we attempt to unmix three SAR sources images, which were obtained by mixing gray images by a random mixing matrix [14]. The SAR images are 256 by 256 pixels, and they are stored as vectors by putting the rows of pixels (i.e., their grayscale values) next to each other. After the mixing we have a 3 by 65536 matrix to which ICA can be applied. The three source SAR images shown in Fig. 4(a) have been mixed using the mixing matrix whose rows are $a_1 = [0.877 \quad 0.779 \quad 0.679]$, $a_2 = [0.013 \quad 0.307 \quad 0.074]$ and $a_3 = [0.310 \quad 0.923 \quad 0.071]$.



(a) The three source SAR images

(b) The separated images using ERICA

(c) The separated images using RICANN

Fig. 4. The separation example Ⅰ

We use the RICANN introduced in section 3 to separate mixed images. Fig. 3 shows the score diagnostic plot, in which we display the robust score distance SD_i of each observation on the vertical axis and indexes of observations on the horizontal axis. Now we want to measure whether unmixing matrix W has done a good job. We use the inaccuracy measure proposed in [15]:

$$\mathrm{INACC} = \frac{\sum_{i=1}^{N}\left(\sum_{j=1}^{N}\frac{|\mathbf{Q}_{ij}|}{\max_k |\mathbf{Q}_{ik}|}-1\right)+\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\frac{|\mathbf{Q}_{ij}|}{\max_k |\mathbf{Q}_{kj}|}-1\right)}{2N(N-1)} \tag{12}$$

where $\mathbf{Q} = WA = (\mathbf{Q}_{ij})_{i,j=1,\cdots,m}$. In the ideal case $\mathbf{Q}$ $\psi$is the product of a permutation matrix and a diagonal matrix with diagonal entries 1 and -1, that is, a matrix which has mostly zeros except for a single nonzero value, either 1 or -1, in each row and in each column. In that case INACC = 0. At the other extreme, the worst case is when all $|\mathbf{Q}_{ij}|$ are equal, and then INACC = 1.

In our paper, the ICA algorithm used in RICANN is ERICA, so we compared our method with ERICA [5]. Fig. 4 shows the resulting separation images using our method and ERICA, respectively. In the figure it can be seen how the separation results are clearly improved by our method. The INACC for the separation images of using ERICA are found to be 34.86%. But the INACC for the separation images of use our method with reject rules is found to be 18.23%.

4.2 Case Ⅱ

In this example, we add 10 dB Gaussian white noise to the three SAR images. The images have been mixed using the mixing matrix whose rows are $a_1 = [0.301 \quad 0.698 \quad 0.854]$, $a_2 = [0.542 \quad 0.378 \quad 0.594]$ and $a_3 = [0.151 \quad 0.860 \quad 0.497]$. Fig. 5 shows the score diagnostic plot. The SAR image separation results are showed in Fig. 6. The INACC for the separation images of using ERICA are found to be 26.33%. But the INACC for the separation images of use our method with reject rules is found to be 8.12%.

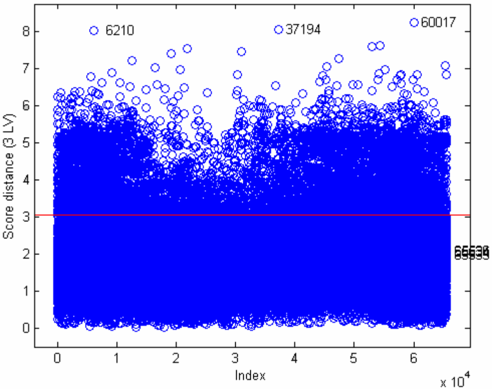
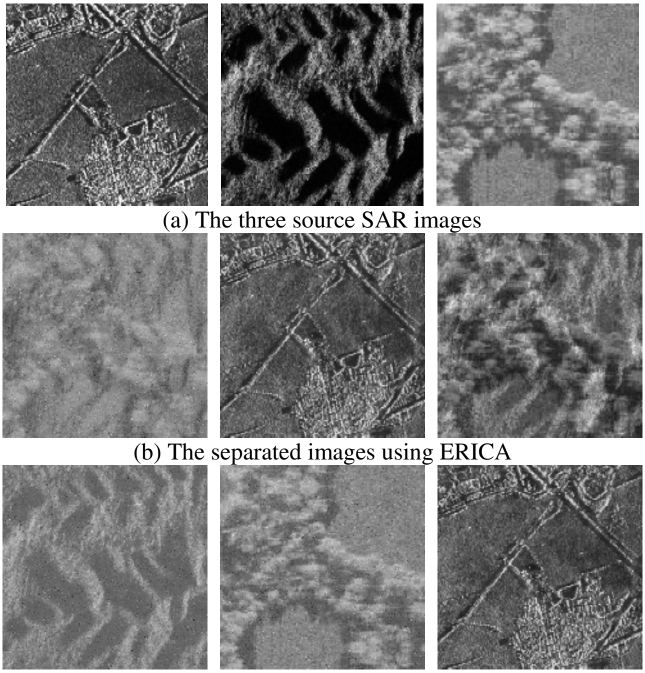


Fig. 5. The score diagnostic plot of example Ⅱ

5 Experiments with Multi-frequency Polarimetric SAR Image Enhancement and Feature Extraction

Recent advances in SAR with multiple frequencies and polarizations, such as those developed by NASNJPL, provide a rich set of data for the same scene. The amount of information is scattered in many images that are correlated as indicated by the high correlation coefficients. The ability of the feature extraction methods to pack



(a) The three source SAR images

(b) The separated images using ERICA

(c) The separated images using RICANN

Fig. 6. The separation example Ⅱ

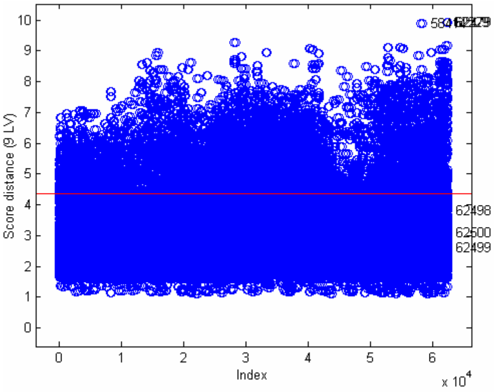
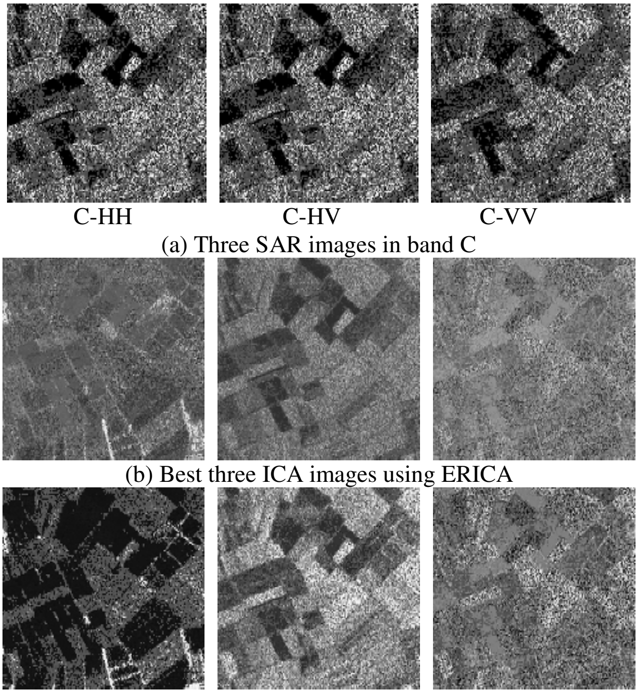


Fig. 7. The score diagnostic plot of multi-frequency polarimetric SAR image

information, decorrelate images, and reduce the noise enables efficient automated image classification and better human scene interpretation.

The considered image data is from an agricultural area near the village of Feltwell, United Kingdom and consists of 9 channels of SAR images: The data consist of three frequency bands; in each band there are three different polarizations (HH, HV, and VV); the available SAR images in polarizations HH, HV and VV of band C, are depicted in the Fig. 8 (a). Figure 7 shows the score diagnostic plot. Fig. 8(b) and (c) show the best three ICA image, produced by the proposed ERICA and RICANN, respectively. The RICANN images are better in contrast than the ERICA image. The RICANN appears as promising model for multi-frequency polarimetric SAR image analysis and interpretation.



(a) Three SAR images in band C

(b) Best three ICA images using ERICA

(c) Best three ICA images using RICANN

Fig. 8. Enhancement result for multi-frequency polarimetric SAR image

6 Conclusion

In this paper, we have proposed a new robust independent component neural network for SAR images analysis in the presence of speckle noise. The method tries to overcome the limitations that the ICA methods have in this kind of signals. The method proposed here is to preprocess the data by rejecting outliers based on orthogonal distance and score distance outlyingness measure, using a high enough cutoff value. We show how our method can be used respectively for blind SAR image separation and for multi-frequency polarimetric SAR image enhancement.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (60375003) , Aeronautics and Astronautics Basal Science Foundation of China (03I53059).

References

1. Bell, A.J. and Sejnowski, T.J.: An information maximization approach to blind separation and blind deconvolution. Neural Computation. vol. 7 (1999)1129-1159.
2. Lee, T.; Lewicki, M. and Sejnowski, T.: Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources. Neural Computation. vol. 11(1999)409-433.
3. Cardoso, J.-F.: High-order contrasts for independent component analysis. Neural Computation. vol. 11(1999)157-192.
4. Amari, S. and Cichocki, A.: Adaptive blind signal processing - neural network approaches. Proc. IEEE. vol. 86(1998)2026-2048.
5. Cruces, S.; Castedo, L.; Cichocki, A.: Robust blind source separation algorithms using cumulants. Neurocomputing. vol. 49, Dec. (2002) 87-118.
6. Hyvarinen, A.: Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. IEEE Trans. NN. vol.10(1999)626-634.
7. Hyvarinen, A.: Gaussian moments for noisy independent component analysis. IEEE Signal Processing Letters. vol. 6(1999)145-147.
8. Attias, H.: Independent Factor Analysis. Neural Computation. vol. 11(1999)803-851.
9. Moulines, E.; Cardoso, J. F.; Gassiat, E.: Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. Proc. ICASSP'97. vol. 5(1997) 3617-3620.
10. Santosh Pandey, Nedret Billor and Asuman Turkmen: The effect of outliers in independent component analysis. Twelfth Annual International Conference on Statistics, Combinatorics, Mathematics and Applications. December, 2005.
11. Hubert, M. Rousseeuw, P.J. Vanden Branden, K.: ROBPCA: a new approach to robust principal component analysis. Technometrics. 2005(47) 64–79.
12. Rousseeuw, P.J., and Van Driessen, K.: A Fast Algorithm for the Minimum Covariance Determinant Estimator. Technometrics. 1999(41), 212–223.
13. Adams, J. B. Smith, M. O. and Johnson, P. E.: Spectral mixture modeling-A new analysis of rock and soil types at the Viking Lander 1 site. *J. Geophys. Res.* vol. 91(B8), (1986) 8090–8112.
14. Cichocki, A. Amari, S.: Adaptive Blind Signal and Image Processing. John Wiley and Sons, Cichester (2002).
15. Giannakopoulos,X.;Karhunen, J.; Oja E.: An experimental comparison of neural algorithms for independent component analysis and blind separation. International Journal of Neural Systems. 1999, 9(2): 99–114.

Kernel Uncorrelated Discriminant Analysis for Radar Target Recognition

Ling Wang, Liefeng Bo, and Licheng Jiao

Institute of Intelligent Information Processing
710071, Xidian University, Xi'an, China
{wliiip, blf0218}@163.com

Abstract. Kernel fisher discriminant analysis (KFDA) has received extensive study in recent years as a dimensionality reduction technique. KFDA always encounters an intrinsic singularity of scatter matrices in the feature space, namely 'small sample size' (SSS) problem. Several novel methods have been proposed to cope with this problem. In this paper, kernel uncorrelated discriminant analysis (KUDA) is proposed, which not only can bear on the SSS problem but also extract uncorrelated features, a desirable property for many applications. And then, we have conducted a comparative study on the application of KUDA and other variants of KFDA in radar target recognition problem. The experimental results indicate the effectiveness of KUDA and illustrate the utility of KFDA on the problem.

1 Introduction

Radar target recognition is a difficulty of task in pattern recognition due to the complex movement of radar target, including transformation and rotation. Particularly for military application, the target is so incooperative that the samples data is much insufficient and noisy. A very simple and rapid approach for recognizing radar target is through the use of radar range profiles which are essentially one-dimension radar images. Due to the high dimensionality of range profiles, it is necessary to perform feature extraction at first to reduce the dimensionality and then perform classification for recognition.

Linear discriminant analysis (LDA), also called fisher discriminant analysis is a widely-used statistical method for feature extraction and dimension reduction, which has been successfully applied in many problems such as face recognition. Because of the nature of linearity, LDA is inadequate to describe the complexity in real world problems. The nonlinearly clustered structure is not easily captured by LDA. In recent years, a category of nonlinear algorithms using the so-called kernel trick have aroused considerable interest in the fields of pattern recognition and machine learning [1]. Generalization of LDA for solving nonlinear problems based on kernel trick has become an active research area. A group of kernel-based fisher discriminant analysis (KFDA) algorithms has been proposed [2]. Extensive empirical comparisons have shown that KFDA works as well as other kernel based classifiers. However, because

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 404 – 411, 2006.

of the implicit high-dimensional nonlinear mapping, the so-called "small sample size" (SSS) problem is very common in the feature space.

Several techniques that might alleviate this problem have been proposed. Mika et al. used the regularization technique to make the inner product matrix nonsingular [3]. But his method was developed to handle binary classification only. Following that, Baudat and Anouar developed a GDA for multiple classification [4]. Yang et al. performed LDA in KPCA feature space to deal with the problem [5]. Recently, Park et al. proposed a kernel based disciminant analysis based on the generalized singular value decomposition called KDA/GSVD, which works regardless of the nonsingularity of the scatter matrices in either the input space or feature space [6].

For feature extraction, the uncorrelated attributes with minimum redundancy are highly desirable. Jin et al. proposed uncorrelated LDA (ULDA) for extracting feature vectors with uncorrelated attributes [7]. However, the proposed method has two limitations, i.e. the expensive computation of the *d* generalized eigenvalue problems, where *d* is number of optimal discriminant vectors by ULDA, and the non-applicability to the SSS problem as the classical LDA. To overcome these limitations, Ye et al. presented an efficient algorithm to compute the optimal discriminant vectors of ULDA and at the same time addressed the SSS problem of ULDA [8]. In [9], the optimization criteria of classical LDA was extended to solve the SSS problem, and the solutions to the proposed criterion form a family of algorithms to which ULDA and a novel algorithm, namely orthogonal LDA (OLDA) belong.

In this paper, we present the nonlinear extension of ULDA based on kernel trick, called KUDA, which can work regardless of the SSS problem. We also investigate the application of KUDA and some KFDA variants in radar target recognition problem. Through the experiments, we not only demonstrate that KUDA is an effective nonlinear dimension reduction approach, but also conclude that all the KFDA variants achieve higher classification accuracy on radar target recognition problem compared with classical LDA. Another surprisingly observation is that a special kernel function, Cauchy kernel, has a remarkable performance on the problem.

2 Related Work on Kernel Fisher Discriminant Analysis

Classical fisher discriminant analysis aims to find the optimal transformation, which maximizes the between-class scatter matrix while minimizing the within-class scatter matrix simultaneously. Thus, the cluster structure of the original high-dimensional space is preserved in the reduced-dimensional space. But this method fails for a nonlinear problem. There have been extensive researches in nonlinear discriminant analysis using kernel function, called by a joined name kernel fisher discriminant analysis (KFDA). Due to the nonlinear map by a kernel function, the dimension of the feature space often becomes much larger than that of the original data space, and as a result, the scatter matrices become singular, which is referred to as "small sample size" (SSS) problem. In the following, we will review some recent proposed KFDA algorithms, all of which attempt to deal with the SSS problem in the feature space.

KPCA plus LDA. PCA plus LDA, a two stage approach, is a popular technique for face recognition [5]. In Euclidean space, the theoretical foundation of why LDA can be performed in the PCA transformed space has been given in [10]. Since real-world problems are always turned into SSS problems by a nonlinear mapping, we can generalize the result directly to the data in a mapped feature space. At first stage, PCA is performed in the feature space. It is equivalent to performing KPCA in the input space. And then, in the KPCA transformed space, LDA is performed.

The biggest challenge in using KPCA plus LDA is that it is difficult to choose an optimal reduced dimension m. If m is chosen large, the eigenvalue problem in the discriminant stage will be expensive and unstable because of the high dimensionality. If too small, it may not provide sufficient discriminant information.

GDA. Generalized discriminant analysis (GDA) is proposed for multiclass classification. As such for LDA, the purpose of GDA method is to maximize the between class scatter matrix while minimizing the within class scatter matrix in the feature space. In order to cope with the singularity of scatter matrices in the feature space, the eigenvectors decomposition of the kernel matrix is employed, and the singularity is avoided by removing some small eigenvalues. As KPCA plus LDA, it is difficult to determine the magnitude of eigenvalue that should be removed.

KDA/GSVD. A recent work on overcoming SSS problem in LDA lies in the use of Generalized Singular Value Decomposition (GSVD), named LDA/GSVD [11]. The method avoids inversing the within-class scatter matrix, so it computes the solution exactly without losing any information. Recently, Park presented the nonlinear extension of LDA based on kernel functions and the GSVD, named KDA/GSVD. The GSVD is employed to solve the generalized eigenvalue problem which is formulated in the feature space defined by a nonlinear mapping through kernel functions. The adventage of KDA/GSVD is that it can be applied regardless of singularity of the scatter matrics both in the original space and in the feature space. The detailed derivation can be found in [6].

3 Kernel Uncorrelated Discriminant Analysis

Uncorrelated linear discriminant analysis (ULDA) [7] was proposed for feature extraction. The feature vectors transformed by ULDA were shown to be statistically uncorrelated, which is a desirable property for many applications. ULDA aims to find the optimal discriminant vectors that are S_t-orthogonal (Two vectors x and y are S_t-orthogonal, if $x^T S_t y = 0$). In this section, we present a nonlinear extension of ULDA based on kernel functions, and solve it using the technique of simultaneous diagonalization of the three scatter matrices [9].

Let n denotes the dimension of the original sample space, and r is the number of classes. And let $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_l\}$ be the training samples set, where $\mathbf{x}_i \in X \subset R^n$. For a given nonlinear mapping ϕ, the input data space R^n can be mapped into the feature space F: $\phi : R^n \to F$. As a result, a sample in the original input space R^n is mapped into a potentially much higher dimensional feature vector: $\mathbf{x} \to \phi(\mathbf{x})$ in the feature space F . To avoid computing the dot products in a high-dimensional feature space, kernel trick is introduced to facilitate the computation. A kernel is defined by an inner product $k(\mathbf{x}_i, \mathbf{x}_j) = \left(\phi(\mathbf{x}_i) \bullet \phi(\mathbf{x}_j)\right)$.

Let $\mathbf{K} = \left[k(\mathbf{x}_i, \mathbf{x}_j)\right]_{(1\le i\le l, 1\le j\le l)}$ be the kernel matrix. Then, we can consider each column in $\mathbf{K}$ as a data point in the n–dimensional space. As in the LDA, we define between-class scatter matrix and within-class scatter and total scatter matrix in the feature space as below:

$$\mathbf{S}_b^F = \mathbf{K}_b \mathbf{K}_b^T , \ \mathbf{S}_w^F = \mathbf{K}_w \mathbf{K}_w^T , \ \mathbf{S}_t^F = \mathbf{K}_t \mathbf{K}_t^T \ , \tag{1}$$

where

$$\begin{aligned}
&\mathbf{K}_b = [b_{ij}]_{(1\le i\le n, 1\le j\le r)}, \quad b_{ij} = \sqrt{n_j}\left(\frac{1}{n_j}\sum_{s\in N_j} k(\mathbf{x}_i, \mathbf{x}_s) - \frac{1}{n}\sum_{s=1}^{n} k(\mathbf{x}_i, \mathbf{x}_s)\right) \\
&\mathbf{K}_w = [w_{ij}]_{(1\le i\le n, 1\le j\le n)}, \quad w_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{n_l}\sum_{s\in N_l} k(\mathbf{x}_i, \mathbf{x}_s), \ x_j \in \text{class } l\ . \\
&\mathbf{K}_t = [t_{ij}]_{(1\le i\le n, 1\le j\le n)}, \quad t_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{n}\sum_{s=1}^{n} k(\mathbf{x}_i, \mathbf{x}_s)
\end{aligned} \tag{2}$$

According to the definition of S_t-orthogonal discriminant vector, we can define a trace optimization problem in the feature space as follows:

$$\mathbf{G} = \arg \max_{\mathbf{G}\in\mathbb{R}^{p\times t}:\mathbf{G}^T\mathbf{S}_t^F\mathbf{G}=I_t} \left(trace(\mathbf{G}^T\mathbf{S}_w^F\mathbf{G})^{-1}\mathbf{G}^T\mathbf{S}_b^F\mathbf{G}\right). \tag{3}$$

Since $\mathbf{S}_t^F = \mathbf{S}_w^F + \mathbf{S}_b^F$, the problem above is equivalent to

$$\mathbf{G} = \arg \max_{\mathbf{G}\in\mathbb{R}^{p\times t}:\mathbf{G}^T\mathbf{S}_t^F\mathbf{G}=I_t} \left(trace(\mathbf{G}^T\mathbf{S}_t^F\mathbf{G})^{-1}\mathbf{G}^T\mathbf{S}_b^F\mathbf{G}\right). \tag{4}$$

Note that $\mathbf{S}_t^F$ and $\mathbf{S}_b^F$ are both singular. In order to solve the problem, a natural extension is that the inverse of a matrix is replaced by the pseudo-inverse [12]:

$$\mathbf{G} = \arg \max_{\mathbf{G}\in\mathbb{R}^{p\times t}:\mathbf{G}^T\mathbf{S}_t^F\mathbf{G}=I_t} \left(trace(\mathbf{G}^T\mathbf{S}_t^F\mathbf{G})^{+}\mathbf{G}^T\mathbf{S}_b^F\mathbf{G}\right). \tag{5}$$

The above optimization problem can be solved by diagonalizing the three scatter matrices $\mathbf{S}_b^F$, $\mathbf{S}_w^F$, and $\mathbf{S}_t^F$ simultaneously.

Let $\mathbf{K}_t = \mathbf{U\Sigma V}^T$ be the SVD of $\mathbf{K}_t$, where $\mathbf{K}_t$ is defined in (2), $\mathbf{U}$ and $\mathbf{V}$ are orthogonal, $\mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Sigma}_t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$, $\mathbf{\Sigma}_t \in \mathbb{R}^{t\times t}$ is diagonal, and $t = rank(\mathbf{S}_t^F)$. Then, we have

$$\mathbf{S}_t^F = \mathbf{K}_t\mathbf{K}_t^T = \mathbf{U\Sigma V}^T\mathbf{V\Sigma}^T\mathbf{U}^T = \mathbf{U\Sigma\Sigma}^T\mathbf{U}^T = \mathbf{U}\begin{pmatrix}\mathbf{\Sigma}_t^2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\mathbf{U}^T. \tag{6}$$

Let $\mathbf{U} = (\mathbf{U}_1, \mathbf{U}_2)$ be a partition of $\mathbf{U}$, such that $\mathbf{U}_1 \in \mathbb{R}^{n\times t}, \mathbf{U}_2 \in \mathbb{R}^{n\times(n-t)}$. (6) can be rewritten as

$$\begin{aligned}\begin{pmatrix}\mathbf{\Sigma}_t^2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix} &= \mathbf{U}^T(\mathbf{S}_b^F + \mathbf{S}_w^F)\mathbf{U}\\ &= \begin{pmatrix}\mathbf{U}_1^T\\ \mathbf{U}_2^T\end{pmatrix}\mathbf{S}_b^F(\mathbf{U}_1, \mathbf{U}_2) + \begin{pmatrix}\mathbf{U}_1^T\\ \mathbf{U}_2^T\end{pmatrix}\mathbf{S}_w^F(\mathbf{U}_1, \mathbf{U}_2) \qquad .\\ &= \begin{pmatrix}\mathbf{U}_1^T\mathbf{S}_b^F\mathbf{U}_1 & \mathbf{U}_1^T\mathbf{S}_b^F\mathbf{U}_2\\ \mathbf{U}_2^T\mathbf{S}_b^F\mathbf{U}_1 & \mathbf{U}_2^T\mathbf{S}_b^F\mathbf{U}_2\end{pmatrix} + \begin{pmatrix}\mathbf{U}_1^T\mathbf{S}_w^F\mathbf{U}_1 & \mathbf{U}_1^T\mathbf{S}_w^F\mathbf{U}_2\\ \mathbf{U}_2^T\mathbf{S}_w^F\mathbf{U}_1 & \mathbf{U}_2^T\mathbf{S}_w^F\mathbf{U}_2\end{pmatrix}\end{aligned} \tag{7}$$

Since both $\mathbf{S}_b^F$ and $\mathbf{S}_w^F$ are positive semidefinite, we thus have

$$\mathbf{U}^T\mathbf{S}_b^F\mathbf{U} = \begin{pmatrix}\mathbf{U}_1^T\mathbf{S}_b^F\mathbf{U}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}, \ \mathbf{U}^T\mathbf{S}_w^F\mathbf{U} = \begin{pmatrix}\mathbf{U}_1^T\mathbf{S}_w^F\mathbf{U}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}. \tag{8}$$

According to (7) and (8), we can derive the following equation

$$\mathrm{I}_t = \mathbf{\Sigma}_t^{-1}\mathbf{U}_1^T\mathbf{S}_b^F\mathbf{U}_1\mathbf{\Sigma}_t^{-1} + \mathbf{\Sigma}_t^{-1}\mathbf{U}_1^T\mathbf{S}_w^F\mathbf{U}_1\mathbf{\Sigma}_t^{-1}. \tag{9}$$

Denote $\mathbf{B} = \mathbf{\Sigma}_t^{-1}\mathbf{U}_1^T\mathbf{K}_b$ and let $\mathbf{B} = \mathbf{P}\tilde{\mathbf{\Sigma}}\mathbf{Q}^T$ be the SVD of $\mathbf{B}$. Then, we get

$$\mathbf{\Sigma}_t^{-1}\mathbf{U}_1^T\mathbf{S}_b^F\mathbf{U}_1\mathbf{\Sigma}_t^{-1} = \mathbf{P}\tilde{\mathbf{\Sigma}}^2\mathbf{P}^T = \mathbf{P\Sigma}_b\mathbf{P}^T, \tag{10}$$

where $\mathbf{\Sigma}_b \equiv \tilde{\mathbf{\Sigma}}^2 = diag(\lambda_1, \cdots, \lambda_t),\ \lambda_1 \geq \cdots \geq \lambda_q > 0 = \lambda_{q+1} = \cdots = \lambda_t$, and $q = rank(\mathbf{S}_b^F)$. It follows from (9) that

$$\mathbf{P}^T\mathbf{\Sigma}_t^{-1}\mathbf{U}_1^T\mathbf{S}_w^F\mathbf{U}_1\mathbf{\Sigma}_t^{-1}\mathbf{P} = \mathbf{I}_t - \mathbf{\Sigma}_b \equiv \mathbf{\Sigma}_w. \tag{11}$$

According to (9), (10), and (11), $\mathbf{S}_b^F$, $\mathbf{S}_w^F$ and $\mathbf{S}_t^F$ can be diagonalized as

$$\mathbf{X}^T\mathbf{S}_b^F\mathbf{X} = \begin{pmatrix}\mathbf{\Sigma}_b & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix} \equiv \mathbf{D}_b, \mathbf{X}^T\mathbf{S}_w^F\mathbf{X} = \begin{pmatrix}\mathbf{\Sigma}_w & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix} \equiv \mathbf{D}_w, \ \mathbf{X}^T\mathbf{S}_t^F\mathbf{X} = \begin{pmatrix}\mathbf{I}_t & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix} \equiv \mathbf{D}_t \tag{12}$$

where $\mathbf{X} = \mathbf{U}\begin{pmatrix}\mathbf{\Sigma}_t^{-1}\mathbf{P} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{pmatrix}$.

Let $\mathbf{X}_q$ be the matrix consisting of the first q columns of $\mathbf{X}$, where $q = rank(\mathbf{S}_b^F)$. $\mathbf{G}^F = \mathbf{X}_q$ is the solution to the optimization problem (5) [9]. Consequently, kernel uncorrelated discriminant analysis (KUDA) algorithm can be described as the following.

Algorithm. KUDA

```
Given a data matrix $X = [\mathbf{x}_1, \cdots, \mathbf{x}_l] \in \mathbb{R}^{n \times l}$ with $r$ classes and a kernel function $k$
1. Compute $\mathbf{K}_b \in \mathbb{R}^{n \times r}$, $\mathbf{K}_w \in \mathbb{R}^{n \times n}$, and $\mathbf{K}_t \in \mathbb{R}^{n \times n}$ as in (2);
2. Compute the reduced SVD of $\mathbf{K}_t$ as $\mathbf{K}_t = \mathbf{U}_1 \mathbf{\Sigma}_t \mathbf{V}_1^T$;
3. $\mathbf{B} = \mathbf{\Sigma}_t^{-1} \mathbf{U}_1^T \mathbf{K}_b$;
4. Compute SVD of $\mathbf{B}$ as $\mathbf{B} = \mathbf{P} \tilde{\mathbf{\Sigma}} \mathbf{Q}^T$; $q = rank(\mathbf{B})$;
5. $\mathbf{X} = \mathbf{U}_1 \mathbf{\Sigma}_t^{-1} \mathbf{P}$;
6. $\mathbf{G}^F = \mathbf{X}_q$;
```

4 Performance Comparison on Radar Target Recognition

Radar target recognition refers to the detection and recognition of target signatures using high-resolution range profiles, in our case, in inverse synthetic aperture radar. A radar image represents a spatial distribution of microwave reflectivity that is sufficient to characterize the illuminated target. Range resolution allows the sorting of reflected signals on the basis of range. When range-gating or time-delay sorting is used to interrogate the entire range extent of the target space, a one-dimensional image, called a range profile, will be generated.

Our task is to recognize the range profile of the three different plane models, i.e. J-6, J-7 and B-52, based on experimental data acquired in a microwave anechoic chamber. The dimensionality of the range profiles is 64. The full data set is split into 363 training samples and 721 test samples. Training samples consist of 104 1-dimensional images of J-6, 151 1-dimensional images of J-7 and 108 1-dimensional images of B-52. Test samples consist of 207 1-dimension images of J-6, 300 1-dimension images of J-7 and 214 1-dimension images of B-52.

Table 1. Number of misclassification of several classifiers

Method	Recognition Rate	Error Number
LDA	94.73	38
GDA	98.61	10
KPCA-LDA	99.69	2
KDA/GSVD	99.71	2
KUDA	99.86	1

A simple classifier, *k*-nearest neighbor (KNN), is employed to evaluate the quality of different dimension reduction algorithms. Leave-one-out error is used to find the best number of neighbor *k*. The experimental results for several methods using an optimal kernel function are summarized in Table 1. For KPCA-LDA, we find the optimal value of the principle components on an interval. From Table 1, we can see that only one wrong recognition occur in KUDA, and only 2 in KDA/GSVD and KPCA-LDA. This indicates that these algorithms proposed to bear on SSS problem are superior to LDA, GDA, and have similar high performance on the radar target recognition problem.

After performing discriminant analysis, the dimensionality of range files is reduced to 2 because the class number is three. Therefore, these real world data can be visualized in Figure 2. From the projection image of low dimension, we can see that LDA is not good enough because of the intrinsic nonlinearity for the problem, and on the contrary, the variants of kernel based discriminant analysis preserve the information for classification well.

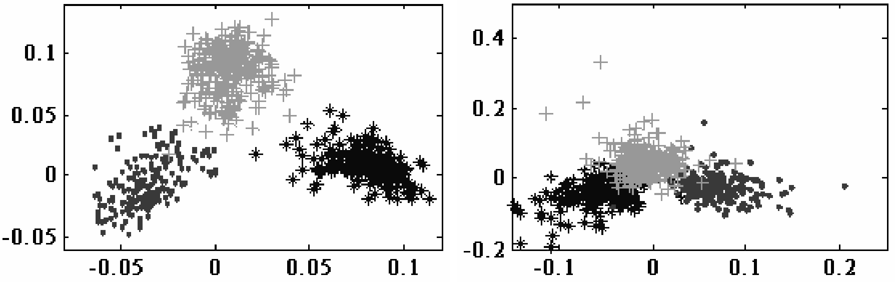


Fig. 2. 2-dimensional visualization of the radar range profiles with kernel (left) and without kernel (right)

Table 2. Performance of variants of kernel discriminant analysis with different kernels

Method	RBF	Coswave	Cauchy
GDA	98.61	98.20	98.61
KPCA-LDA	97.23	97.09	99.69
KDA/GSVD	97.45	97.05	99.71
KUDA	97.23	97.09	99.86

We also compare the performance of variants of kernel fisher discriminant analysis on three popular kernels, i.e. Gaussian RBF kernel $k\left(\mathbf{x},\mathbf{y}\right)=\exp(\frac{-\left\|\mathbf{x}-\mathbf{y}\right\|^2}{2p^2})$, Coswave kernel $k\left(\mathbf{x},\mathbf{y}\right)=\frac{p}{p+\left\|\mathbf{x}-\mathbf{y}\right\|^2}$ and Cauchy kernel $k\left(\mathbf{x},\mathbf{y}\right)=\cos(1.75\times\frac{(\mathbf{x}-\mathbf{y})}{p})\exp(-\frac{\left\|\mathbf{x}-\mathbf{y}\right\|^2}{2p^2})$, where $p\in R$. The results are summarized in Table 2. From the experimental results, we find unexpectedly that the Cauchy kernel has a predominant performance on the problem.

5 Conclusion

In this paper, we propose a new kernel fisher discriminant analysis, namely KUDA to deal with the SSS problem in the feature space. And then, we describe the application of KUDA and some other KFDA variants in radar target recognition problem. Experiment results have shown that KUDA and the KFDA variants developed for solving the SSS problem perform significantly better than the classical LDA. Furthermore, it is worth to mention that a specific kernel, i.e. Cauchy kernel, performs best on the problem. These observations are expected to be useful when we attempt to apply kernel discriminant analysis to other target recognition problems.

References

1. Bo, L.F., Wang, L., and Jiao, L.C.: Training support vector machines using greedy stagewise algorithm. Lecture Notes in Computer Science (PAKDD'05) 3518 (2005) 632-638
2. Bo, L.F., Wang, L., and Jiao, L.C.: Feature scaling for kernel fisher discriminant analysis using leave-one-out cross validation. Neural Computation 18(4) (2006) 961-978
3. Mika, S., Ratsch, G., and Weston, J.: Fisher discriminant analysis with kernels. In Proceedings of the IEEE Workshop on Neural Networks for signal Processing (1999) 41-48
4. Baudat, G. and Anouar, F.: Generalized discriminant analysis using a kernel approach. Nerual Compuation 12(10) (2000) 2385-2404
5. Yang, M.H.: Kernel eigenfaces vs. kernel fisherfaces: face recognition using kernel methods. In Proceedings of Fifth IEEE International Conference Automatic Face and Gesture Recognition (2002) 215-220
6. Park, C.H. and Park, H: Nonlinear discriminant analysis using kernel functions and the generalized singular value decomposition, SIAM Journal on Matrix Analysis and Applications, to appear
7. Jin, Z., Yang, J.Y., Tang, Z.M., and Hu, Z.S.: A theorem on the uncorrelated optimal discriminant vectors. Pattern Recognition 34 (2001) 2041-2047
8. Ye, J.P, Janardan, R., Li, Q., and Park, H.: Feature extraction via generalized uncorrelated linear discriminant analysis. In Proceedings of the 21st International Conference on Machine Learning, Banff, Canada 2004
9. Ye, J.P.: Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. Journal of Machine Learning Research 6(4) (2005) 483-502
10. Yang, J., Yang, J.Y.: Why can LDA be performed in PCA transformed space? Pattern Recognition 36 (2003) 563-566
11. Howland, P. and Park, H.: Generalizing Discriminant Analysis Using the Generalized Singular Value Decomposition. IEEE Transactions on PAMI 26(8) (2004) 995-1006
12. Ye, J.P., Janardan, R., Park, C.H., and Park, H.: An optimization criterion for generalized discriminant analysis on undersampled problems. IEEE Transactions on PAMI 26(8) (2004) 982-994

SuperResolution Image Reconstruction Using a Hybrid Bayesian Approach

Tao Wang, Yan Zhang, and Yong Sheng Zhang

Zhengzhou Institute of Surveying and Mapping, No. 66 Longhai Middle Road,
Zhengzhou 450052, China
wangtaoynl@163.com

Abstract. There are increasing demands for high-resolution (HR) images in various applications. Image superresolution (SR) reconstruction refers to methods that increase image spatial resolution by fusing information from either a sequence of temporal adjacent images or multi-source images from different sensors. In the paper we propose a hybrid Bayesian method for image reconstruction, which firstly estimates the unknown point spread function(PSF) and an approximation for the original ideal image, and then sets up the HMRF image prior model and assesses its tuning parameter using maximum likelihood estimator, finally computes the regularized solution automatically. Hybrid Bayesian estimates computed on actual satellite images and video sequence show dramatic visual and quantitative improvements in comparison with the bilinear interpolation result, the projection onto convex sets (POCS) estimate and Maximum A Posteriori (MAP) estimate.

1 Introduction

There are increasing demands for high-resolution (HR) images in various applications. Although the most direct way to increase spatial resolution is to use a HR image acquisition system, fabrication limitations and high cost for high precision optics and image sensors are always prohibitive concerns in many commercial applications. Therefore, the new image SR reconstruction approach, which is capable of generating a HR image from multiple low-resolution (LR) images, has been a hot research topic recently[1].Since Tsai and Huang's work[2], many work has been reported in the literature, including the weighted least-squares algorithm[1], the nonuniform interpolation approach[1], the POCS method[3-4] and MAP Bayesian approach[5-7]. Among these algorithms, the Bayesian approach is most notable for it robustness and flexibility in modeling noise characteristics and a priori knowledge about the solution. Assuming that the noise process is white Gaussian, the Bayesian estimation with convex energy functions ensures the uniqueness of the solution. But existing Bayesian reconstruction methods suffer from several impractical assumptions. Previous research often assumes that PSF is definitely known during reconstruction, which is impossible for actual images reconstruction as many uncertain blurring factors are involved during imaging process. Further, the image prior model founded upon the upsampled LR image greatly affects the quality of the reconstruction result as the LR images are

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 412 – 419, 2006.

already contaminated and the resulted prior model is not robust to noise. Finally, the edge threshold parameter of the image prior model needs to be adjusted by experienced experts empirically, which limits the wide usage of the Bayesian estimator.

Therefore, we propose a novel hybrid Bayesian estimator for SR image reconstruction. Under the Bayesian framework, it deconvolutes the upsampled LR image to access PSF and approximation value for the ideal HR image with APEX algorithm first, and then models the HMRF image prior model and assesses its edge threshold parameter through maximum likelihood (ML) estimation, finally regularizes the ill-posed reconstruction process automatically.

2 Statement on Hybrid Bayesian Reconstruction Algorithm

Above all we formulate an observation model that relates the original HR image to the observed LR image. Consider the desired HR image $\mathbf{x} =[x_1 , x_2 ,...., x_N]^T$, $N= L_1N_1 \times L_2N_2,$ which is sampled at or above the Nyquist rate from a hypothetically bandlimited continuous scene. L_1 and L_2 are the horizontal and vertical down-sampling factors, respectively. Let the kth LR image be denoted as $\mathbf{y}^{(k)} =[y^{(k)}_1 , y^{(k)}_2 ,...., y^{(k)}_M]^T$, $M = N_1 \times N_2$. During the imaging process, the observed LR image result from warping, blurring, and subsampling operators performed on $\mathbf{x}$ and is also corrupted by additive noise, we can then represent the observation model as

$$\mathbf{y}_k = \mathbf{DHT}_k\mathbf{x} + \mathbf{n}_k = \mathbf{W}_k\mathbf{x} + \mathbf{n}_k \quad \text{for } 1 \le k \le \text{p}. \tag{1}$$

where $\mathbf{T}_k$ is a warp matrix, $\mathbf{H}$ represents a blur matrix **,** $\mathbf{D}$ is a subsampling matrix and $\mathbf{n}_k$ represents a noise vector, assumed to be Gaussian, white and stationary, p is the number of images.

The SR image reconstruction problem is ill-posed. A well-posed problem can be formulated under the MAP stochastic framework by introducing a priori constraint,

$$\mathbf{x} = \arg\max\{\log P(\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_\text{p} | \mathbf{x}) + \log P(\mathbf{x})\}. \tag{2}$$

Both the priori image model $P(\mathbf{x})$ and the conditional density $P(\mathbf{y}_1 ,\mathbf{y}_2 ,\dots ,\mathbf{y}_\text{p} | \mathbf{x})$ will be defined by a priori knowledge concerning $\mathbf{x}$ and the statistical information of noise. If the motion estimation error between images is assumed to be independent and noise is assumed to be an independent identically distributed zero mean Gaussian distribution, the conditional density can be expressed in the compact form

$$P(\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_\text{p} | \mathbf{x}) = \prod_{k=0}^{\text{p}} P(\mathbf{y}_k | \mathbf{x}) = \prod_{k=0}^{\text{p}} \left\{ \frac{1}{(2\pi)^{\frac{N}{2}} \sigma^N} \exp\left\{ -\frac{1}{2\sigma^2} \|\mathbf{y}_k - \mathbf{W}_k\mathbf{x}\|^2 \right\} \right\} . \tag{3}$$

where σ^2 is error variance, $\mathbf{y} =[\mathbf{y}_1 , \mathbf{y}_2 ,...., \mathbf{y}_\text{p}]^T$, $\mathbf{W} =[\mathbf{W}_1 , \mathbf{W}_2 ,...., \mathbf{W}_\text{p}]^T$.

In order to reconstruct the high-frequency information lost through imaging, we take the HMRF prior model, which represents piecewise smooth image data[5],

$$P(\mathbf{x}) = \frac{1}{Z}\exp\{-\frac{1}{2\beta}\sum_{c \in S}\rho(d_l^t\mathbf{x}, \alpha)\} = \frac{1}{Z}\exp\{-\frac{1}{2\beta}\sum_{m,n}\sum_{l=1}^{4}\rho(d_l^t\mathbf{x}, \alpha)\} . \tag{4}$$

where *Z* is a normalizing constraint, β is the temperature parameter, *c* is a local group of pixels contained within the image cliques *S*, α is the edge threshold parameter separating the quadratic and linear regions. The quantity $d_l^t\mathbf{x}$ measures the second-order finite differences in four directions at each pixel in the HR image, small in smooth locations and large at edges[5].The likelihood of edges in the data is controlled by the Huber penalty function

$$\rho(x,\alpha)=\begin{cases} x^2, & |x|\le\alpha, \\ 2\alpha|x|-\alpha^2, & |x|>\alpha, \end{cases}. \tag{5}$$

The regularized solution is then equivalent to minimizing the cost function

$$U(\mathbf{x})=\|\mathbf{y}-\mathbf{W}\mathbf{x}\|^2+\sum_{m,n}\sum_{l=1}^{4}\rho(d_l^t\mathbf{x},\alpha)\cdot \tag{6}$$

The HMRF prior model should be founded on the ideal HR image and parameter α should also be decided upon it. But in the existing MAP research, the upsampling LR image is usually taken as substitute for the ideal HR image and α is set empirically. However, the ideal HR image can't be approximated by its degraded version because the LR image is blurred and noisy. Parameter α estimated on a blurred image has too high a value and leads to over-smoothed solutions. Parameter α estimated on a noisy image is too low, and provides insufficient regularization, leading to noisy solutions. A bad initialization for the prior model often leads to degenerated solutions[7]. The Bayesian estimator is only significant and supplies good regularized estimate in the case of ideal HR image Therefore, an approximation of **x** has to be accurately determined before reconstruction.

We choose APEX algorithm to compute the approximation of **x** as the deconvoluted result produced by APEX algorithm is sufficiently close to the original image to enable us to set up an accurate HRMF prior model. Moreover, the unknown PSF can also be determined. In the following, we detail how to get an approximation of **x** with APEX algorithm, how to estimate parameter α from the approximation image, and how to generate a reconstruction estimate automatically.

3 Hybrid Bayesian Reconstruction Solution

3.1 APEX Prior Blind Deconvolution

The APEX[8] method is a FFT-based direct blind deconvolution technique, which is applicable to a restricted two-dimensional radially symmetric shift-invariant G class blurs. The OTF (Optical Transfer Function) form of G class blur *h*(*x*, *y*) is defined as

$$H(\varepsilon,\eta)=\int_{R^2}h(x,y)e^{-i2\pi(\varepsilon x+\eta y)}dxdy = e^{-a(\varepsilon^2+\eta^2)^b}. \tag{7}$$

where (*a*>0, 0<b<1). When just blurring factor considered, the relationship between the HR image *x*(*x*,*y*) and the LR image *y*(*x*,*y*) in the frequency domain is as follows,

$$Y(\varepsilon,\eta)=H(\varepsilon,\eta)X(\varepsilon,\eta)+N(\varepsilon,\eta)\cdot \tag{8}$$

where $Y(\varepsilon, \eta)$, $X(\varepsilon, \eta)$ and $N(\varepsilon, \eta)$ are Fourier transforms of $x(x, y)$, $y(x, y)$ and $n(x, y)$, respectively. We may surely assume that the noise n(x, y) satisfies $\int_{R^2} |n(x, y)| dxdy \leq f(x, y) dxdy = \sigma > 0$ (σ is a normalizing constant), so that we can ignore N(ε, η) and further normalize (8) into (9), assuming Y(ε, η), X(ε, η) and the OTF keep the following relation in a region Ω in the frequency domain

$$\log|Y(\varepsilon,\eta)| \approx -a(\varepsilon^2 + \eta^2)^b + \log|X(\varepsilon,\eta)| \,. \tag{9}$$

We replace log|$X(\varepsilon,\eta)$| by negative constant -A and solve (a, b) in (9) with nonlinear least squares algorithms. Putting (a, b) into (10), we can get the optimal approximation value for ideal HR image after inverse Fourier transform. $\overline{H}$ is the conjugate of H , K and s are adjustable parameters

$$X(\varepsilon,\eta) = \frac{\overline{H}(\varepsilon,\eta)Y(\varepsilon,\eta)}{|H(\varepsilon,\eta)|^2 + K^{-2}|1 - H^s(\varepsilon,\eta)|^2} \,. \tag{10}$$

3.2 Maximum Likelihood Estimation on HMRF Parameter

The ML estimation of the edge threshold parameter α based on the approximation value provided by APEX deconvolution is calculated as

$$\hat{\alpha} = \arg\max P(\mathbf{x}|\alpha) \cdot \tag{11}$$

Parameter α can be assessed according to a predetermined cutoff ratio T ($T = f_\alpha(d_l^t \mathbf{x}) / f(d_l^t \mathbf{x})$), which corresponds to the percentage of high-frequency components in the image. $f(d_l^t \mathbf{x})$is the norm from($\|\ \|$) of the second order derivative, $f_\alpha(d_l^t \mathbf{x})$ is the norm when α is taken into consider (any value lower than α is set to zero). Since the approximation of the original image is known, T can be chosen according to the available information of energy distribution in the HR image. After ratio T is set, the estimation on α consists of solving the system

$$\partial \log P(\mathbf{x}|\alpha) / \partial\alpha = \partial\left[\sum_r \sum_{l=1}^{4} \rho(d_l^t \mathbf{x}, \alpha)\right] \Big/ \partial\alpha = 0 \cdot \tag{12}$$

where r is the component within the high frequency components set. Thus it gives $\hat{\alpha} = \sum_{r \in R} (|d_l^t \mathbf{x}|) \big/ n$, n is the number of high-frequency components.

3.3 Gradient Projection Solution

We select the improved Newton gradient optimization technique to compute the unique minimum solution, which searches the global minimum of the objective function along the Newton direction. Any starting point $\mathbf{x}_0$ that satisfies (1) is valid. We use APEX restored image as the initial value $\mathbf{x}_0$. Suppose the gradient matrix of the cost function $U(\mathbf{x}_i)$ is $\mathbf{g}_i = \nabla U(\mathbf{x}_i)$ and the Hessian matrix is $\mathbf{G}_i = \nabla^2 U(\mathbf{x}_i)$ (i=0,…,K) , in each iteration the Newton direction $\mathbf{p}_i$ is calculated as

$$\mathbf{p}_i = -\mathbf{G}_i^{-1}\mathbf{g}_i . \tag{13}$$

And $\hat{\mathbf{x}}$ moves in the Newton direction $\mathbf{p}_i$ with step sizeτ_i to minimize $U(\mathbf{x}_i)$.

$$\tau_i = -\frac{\mathbf{p}_i^T \mathbf{p}_i}{\mathbf{p}_i^T (\mathbf{G}_i)\mathbf{p}_i} . \tag{14}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \tau_i \mathbf{p}_i . \tag{15}$$

A sequence of iterates $\{\mathbf{x}_i\}_{i=0}^{K}$, more closely to $\hat{\mathbf{x}}$, are generated. The convergence is achieved until the relative state change for a single iteration has fallen below a predetermined threshold ε, such that $\|\mathbf{x}_{i+1} - \mathbf{x}_i\| / \|\mathbf{x}_i\| \le \varepsilon$.

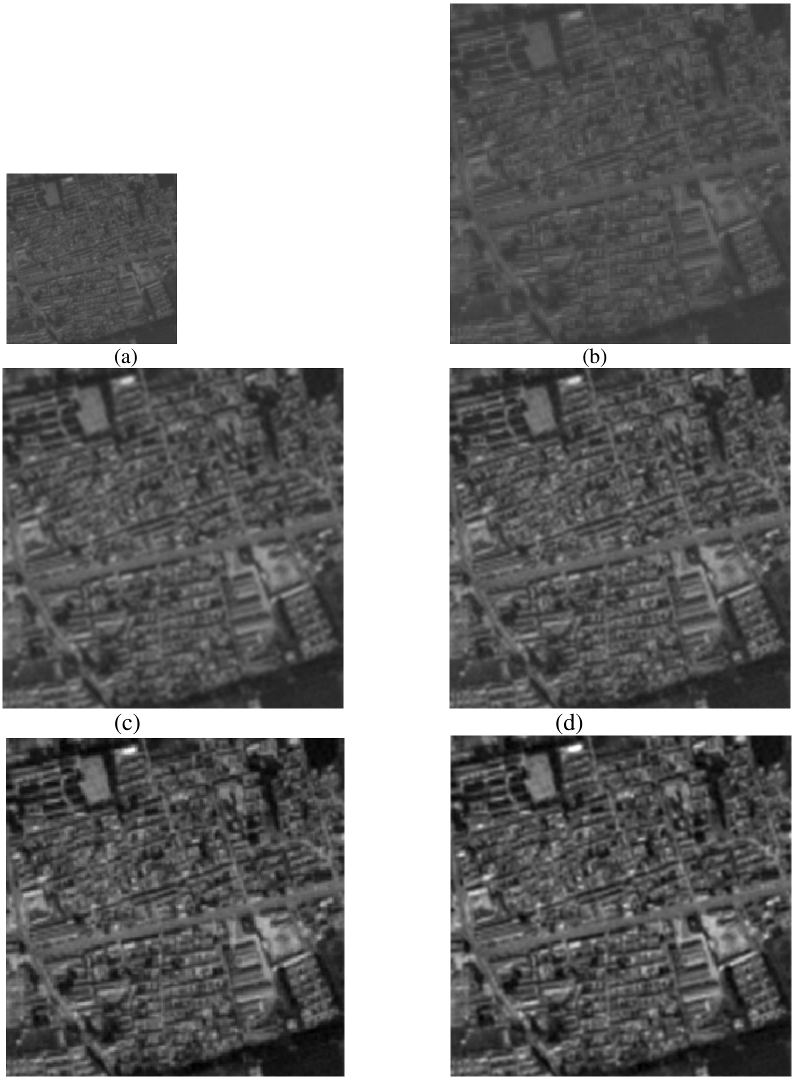
The whole procedure of the hybrid Bayesian estimator is summarized as follows.

1) Upsample the LR images according to the enhancement factor q using bilinear interpolation, construct matrix **D** according to q, construct the geometric distortion matrix **T** using the hierarchical block matching[5].
2) Deconvolute the reference upsampling image with APEX algorithm to obtain the optimal approximation value for HR image and PSF.
3) Calculate the Newton direction $\mathbf{p}_i$.
4) Compute the step size τ_i and update the state according to (14) and (15).
5) If convergence criterion is satisfied, the estimate is given as $\hat{\mathbf{x}} = \mathbf{x}_{i+1}$. Otherwise, increment $\mathbf{x}_{i+1=}$ $\mathbf{x}_i$+$\tau_i\mathbf{p}_i$ and return step 3.

4 Results

In order to demonstrate the performance of the proposed algorithm, two groups of experiment results are presented here, which involve actual satellite remote sensing images and actual video sequence grabbed from a digital video film during play back. The enhancement factor is set to be 2. The bilinear interpolation scheme, the POCS algorithm[3], the Huber-MAP algorithm[5] and our proposed hybrid Bayesian estimator (HBE) are applied in each group of test.

In the first group of test, we try to generate a HR satellite image from a sequence of five 5.0m resolution SPOT 5 satellite images. Fig.1 (a) is the reference 5.0m resolution LR image. The bilinear interpolation of the reference image, the POCS, Huber-MAP and HBE estimates are shown in shown in Fig. 1(b), 1(c), 1(d) and 1(e) respectively. Fig. 1(f) is the 2.5m resolution SPOT 5 image. The PSNR (Peak Signal-to-Noise Ratio) of the bilinear interpolation is 20.1, those of POCS, Huber-MAP and HBE estimates are 24.3, 25.2 and 26.7 respectively. Obviously, the HBE method achieves a significant improvement in PSNR, with considerably much higher resolution than the bilinear interpolation, POCS and Huber-MAP estimates.



(a) (b) (c) (d) (e) (f)

Fig. 1. Actual Satellite Image Sequence. (a) the reference 5.0m image. (b) Bilinear interpolation result. (c) POCS estimate. (d) Huber-MAP estimate. (e) HBE result. (f) the 2.5m HR image.

In the second group of test, nine frames are grabbed from the video sequence during playback. The frame shown in Fig. 2(b) is the bilinear interpolation of the reference frame in Fig. 2(a). The POCS result after 20 iterations is shown Fig. 2(c). The Huber-MAP result after 20 iterations is shown Fig. 2(d) and the HBE result after 16 iterations is shown Fig. 2(e). Fig. 2(f) is the original HR image.

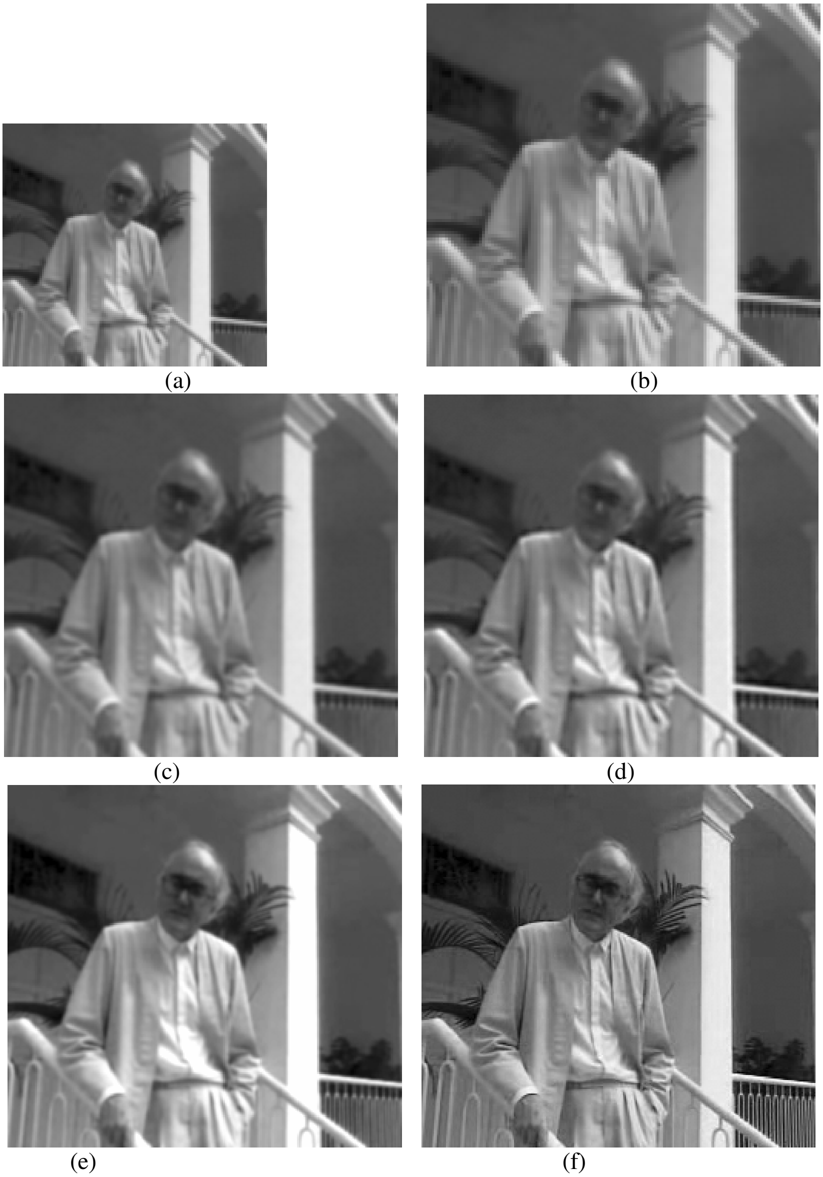


Fig. 2. Actual Video Sequence. (a) the reference LR image. (b) Bilinear interpolation result. (c) POCS estimate. (d) Huber-MAP estimate. (e) HBE result. (f) the HR image.

The PSNR values of the bilinear interpolation, POCS, Huber-MAP and HBE estimates are 22.3, 25.2, 25.7 and 27.1 respectively. Experimental result shows that the image generated by the HBE approach outperforms those produced by bilinear interpolation, POCS and Huber-MAP estimators, especially in the areas of man's face and the bars far behind the man.

5 Conclusion

In the paper a novel hybrid Bayesian algorithm is proposed for HR image reconstruction from actual LR images or video sequence. The proposed approach firstly gets a good approximation of the ideal HR image, then estimate the edge threshold parameter from approximation data by ML estimation, and finally obtains a regularized reconstruction estimate automatically. Its main contributions are setting up an accurate HMRF image prior model, which enables the reconstruction processing to be carried out automatically and ensures the robustness of the estimate. Experimental results demonstrate this new technique is robust and gives very excellent reconstruction result in actual satellite data and video data. The resulted images exhibit much sharper and clearer details than images reconstructed by the bilinear interpolation, the POCS estimator and the Huber-MAP estimator.

References

1. Park, S. C., Park, M. K., Kang, M. G..: Super-Resolution Image Reconstruction: A Technical Overview. IEEE Signal Processing Magazine. 5 (2003) 21-36
2. Tsai R. Y., Huang, T.S.: Multiframe image restoration and registration. In: in Huang, T.S.(Ed.): Advances in computer vision and image processing, JAI Press, (1984) 317-339
3. Patti, A.J., Sezan, M. I., Tekalp, A. M.: Superresolution Video Reconstruction with Arbitrary Sampling Lattices and Nonzereo Aperture Time. IEEE Trans. Image Processing. 8 (1997) 1064-1997
4. Patti, A.J., Altunbasak, Y.: Artifact reduction for set theoretic super resolution image reconstruction with edge adaptive constraints and higher-order interpolants. IEEE Trans. Image Processing. 1(2001) 179-186
5. Schulz, R.R., Stevenson, R. L.: Extraction of High-Resolution Frames from Video Sequences. IEEE Trans. Image Processing. 6 (1996) 996-1011
6. Hardie, R.C., Barnard ,K.J., Armstrong E.E.: Joint MAP registration and high-resolution image estimation using a sequence of undersampled images. IEEE Trans. Image Processing. 12 (1997) 1621-1633
7. Jalobeanu, A., Blanc-Féraud, L. et al.: An Adaptive Gaussian Model for Satellite Image deblurring. IEEE Trans. Image Processing, 4(2004) 613-621
8. CARASSO, A.S.,: THE APEX Method in Image Sharpening and the use of low exponent Lévy Stable Laws. SIAM J. APPL. MATH., 2(2002) 593-618

Retrieval-Aware Image Compression, Its Format and Viewer Based Upon Learned Bases

Naoto Katsumata[1], Yasuo Matsuyama[2], Takeshi Chikagawa[3],
Fuminori Ohashi[2], Fumiaki Horiike[2],
Shun'ichi Honma[2], and Tomohiro Nakamura[2]

[1] Yahoo Japan,
[2] Waseda University, Tokyo 169-8555, Japan
[3] Nomura Research Institute, Japan
{katsu, yasuo, take-c-chika, fumi, fmi_h, shunichi1029,
nt_naka}@wiz.cs.waseda.ac.jp
http://www.wiz.cs.waseda.ac.jp/index-e.html

Abstract. A retrieval-aware image format (rim format) is developed for the usage in the similar-image retrieval. The format is based on PCA and ICA which can compress source images with an equivalent or often better rate-distortion than JPEG. Besides the data compression, the learned PCA/ICA bases are utilized in the similar-image retrieval since they reflect each source image's local patterns. Following the format presentation, an image search viewer for network environments (Wisvi; Waseda image search viewer) is presented. Therein, each query is an image per se. The Wisvi system based on the "rim" method successfully finds similar-images from non-uniform network environments. Experiments support that the PCA/ICA methods are viable to the joint compression and retrieval of digital images. Interested test users can download a β-version of the tool for the joint image compression and retrieval from a web site specified in this paper.

1 Joint Data Compression and Retrieval of Images

Growing popularity of the Internet increases the necessity of image retrieval systems more and more. For instance, the service of flickr [1] helps image sharing among blog groups. Thus, retrieved images migrate frequently among the network environments which contain PC's and mobile phones. In such cases, the direct retrieval from a query image is desirable. Then, computational intelligence methods with learning are expected to contribute to this class of problems Therefore, this paper addresses the following problems:

(a) To utilize learned image bases from the principal component analysis (PCA) and the independent component analysis (ICA) so that the joint data compression and retrieval is effectively achieved. The data compression presented in the text can outperform JPEG. On the similar-image retrieval, the authors had made extensive experiments to compare the color bin method and the learned bases method [2]. Due to this, the main purpose of this paper is

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 420–429, 2006.

set to find the method to achieve the joint performance of the compression and retrieval of digital images.

(b) To define the retrieval-aware image format, say rim (Retrieval-aware IMage format).

(c) To give a useful viewer. As will be observed in the main text, items (a) and (b) are successfully fulfilled. Therefore, systems which can handle images directly as queries become worthy to build. The Wisvi (Waseda Image Searchable VIewer) is presented for this purpose so that uninitiated users can find desirable images via human-friendly method. This system is helpful for the opinion test to measure the system performance.

(d) To design the whole methods and systems to be applicable to scattered network environments as well as databases.

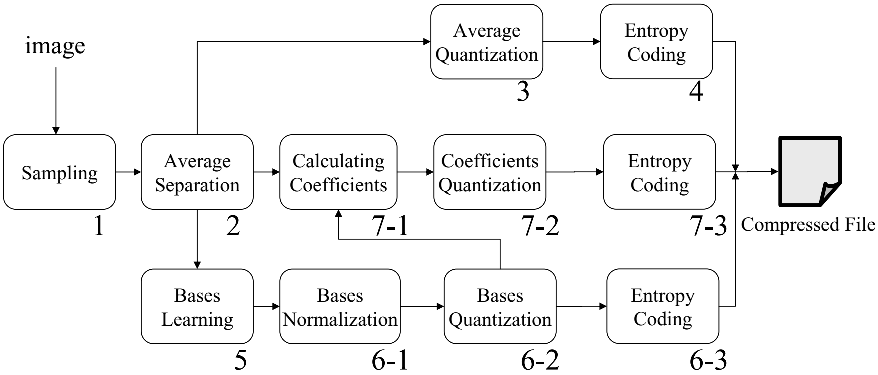


Fig. 1. Retrieval-aware image compression

2 Utilization of Learned Image Bases

As was stated in Section 1, the purpose of this paper is

(a) to show efficient and retrieval-aware image compression methods,
(b) to give an effective format for this purpose,
(c) and to design a user-friendly viewer system.

Figure 1 illustrates the total system for the image compression with the purpose of the similar image retrieval. This system will use PCA and ICA bases after the mean value separation. The blocks with numbers in this figure have the functions explained below. Each item number corresponds to the block number.

(1) Sampling:
At this stage, patches $\{I(x, y)\}$ with the size of $m \times m$ are obtained. Each patch is considered as a vector $\boldsymbol{x}$.

$$\begin{aligned}\boldsymbol{x} &= [\ R(x_1,y_1), R(x_2,y_1),\cdots,R(x_m,y_m),\\ &\quad\ G(x_1,y_1), G(x_2,y_1),\cdots,G(x_m,y_m),\\ &\quad\ B(x_1,y_1), B(x_2,y_1),\cdots,B(x_m,y_m)\ ]^T\\ &\overset{\text{def}}{=} [\boldsymbol{x}_R,\boldsymbol{x}_G,\boldsymbol{x}_B]^T \end{aligned} \tag{1}$$

(2) Average separation:
Color component's sample mean values are computed and subtracted from each component of {R, G, B}.

$$\boldsymbol{x} \leftarrow [\boldsymbol{x}_R-\mu_R,\ \boldsymbol{x}_G-\mu_G,\ \boldsymbol{x}_B-\mu_B]^T \tag{2}$$

In later experiments, the image compression using

$$\boldsymbol{\mu} = [\mu_R,\ \mu_G,\ \mu_B] \tag{3}$$

will be found better than using $\boldsymbol{\mu}_{all-color}$, which is a single vector mean of vector patches, contrary to our naive intuition.

(3) Average quantization:
The average μ_{color} (the index "*color*" stands for R, G, or B) is quantized as follows.

$$\hat{\mu}_{color} \leftarrow \lfloor \mu_{color}/q_{avg} \rfloor \tag{4}$$

The quantization step size is as follows.

$$q_{avg} \leftarrow \lfloor q_{cff}/(1.5m) \rfloor \tag{5}$$

Here, q_{cff} is the quantization size for basis coefficients explained in later sections.

(4) Average entropy coding:
This step computes the difference between contiguous frames as is adopted in JPEG.

$$\Delta\hat{\mu}_{color}(k) \leftarrow \hat{\mu}_{color}(k) - \hat{\mu}_{color}(k-1) \tag{6}$$

After this step, the run-length Huffman coding is executed for the average compression.

(5) PCA bases learning:
Computing the bases for PCA starts from the normalization for the zero mean. This is already completed in the average separation. Then, the covariance is computed by

$$\boldsymbol{C} = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T]. \tag{7}$$

Then, the data reduction matrix is computed.

$$\boldsymbol{V} = \boldsymbol{D}^{-1/2}\boldsymbol{E}^T \tag{8}$$

Here, $\boldsymbol{D}$ is a diagonal matrix of the first L large eigenvalues of $\boldsymbol{C}$. $\boldsymbol{E}$ is a matrix whose columns are eigenvectors corresponding to $\boldsymbol{D}$. Then, the reduced or low-pass filtered vector is expressed by

$$\boldsymbol{z} = \boldsymbol{V}\boldsymbol{x}. \tag{9}$$

Then,

$$\bar{\boldsymbol{x}} \overset{\text{def}}{=} \boldsymbol{V}^{-1}\boldsymbol{z} \overset{\text{def}}{=} \hat{\boldsymbol{U}}_{PCA}\boldsymbol{z} \tag{10}$$

is the image restoration. This $\hat{\boldsymbol{U}}_{PCA}$ is the set of the PCA bases.

(5') ICA bases learning:
After obtaining the PCA bases, another set of powerful bases can be obtained. These are ICA bases (independent component analysis) [3], [4].

$$\hat{\boldsymbol{s}} = \hat{\boldsymbol{W}}\boldsymbol{z} = \hat{\boldsymbol{W}}\boldsymbol{V}\boldsymbol{x} \tag{11}$$

Here, $\hat{\boldsymbol{s}}$ is the estimated coefficients whose components are independent each other. The image restoration is performed by

$$\bar{\boldsymbol{x}} \overset{\text{def}}{=} (\hat{\boldsymbol{W}}\boldsymbol{V})^{-1}\hat{\boldsymbol{s}} \overset{\text{def}}{=} \hat{\boldsymbol{U}}_{ICA}\hat{\boldsymbol{s}}. \tag{12}$$

This $\hat{\boldsymbol{U}}_{ICA}$ is the set of the ICA bases.

(6) Bases normalization (6-1):
Let $\boldsymbol{A}$ be the PCA or ICA basis matrix. First, each basis which is a column vector of $\boldsymbol{A}$ is normalized so that the basis norm is unity: $\|\boldsymbol{a}_i\| = 1$.
Basis quantization (6-2):
Then, the quantization step size is computed.

$$q_{bases}(i) \leftarrow \left\lfloor \left(2^{b_{prec}-1} - 1\right) / a_{max}(i) \right\rfloor \tag{13}$$

Here, $a_{max}(i)$ is the maximum value of the normalized basis $\boldsymbol{a}(i)$. The number b_{prec} sets the granularity of the quantization. Experimentally decided value is 6 bits, i.e., $b_{prec} = 6$. Then, the quantization for the bases is performed by

$$\hat{\boldsymbol{a}}(i) \leftarrow \lfloor \boldsymbol{a}(i) q_{bases}(i) \rfloor. \tag{14}$$

Entropy coding (6-3):
The loss-less data compression is performed by the run-length Huffman coding after computing the difference as is illustrated in Figure 2.

(7) Coefficients calculation (7-1):
Using the quantized bases, the superposition coefficients for the bases are computed.

$$\boldsymbol{s} \leftarrow \boldsymbol{A}^{-1}\boldsymbol{x} \tag{15}$$

Coefficients quantization (7-2):
The i-th component is quantized as follows.

$$\hat{s}(i) \leftarrow \lfloor (s(i) q_{bases}(i)) / q_{cff} \rfloor \tag{16}$$

Here, q_{cff} is a design parameter which can be set by users.
Entropy coding (7-3):
Finally, the run-length Huffman coding is applied column-wise to the coefficient matrix $\boldsymbol{S}$ for the handled image.

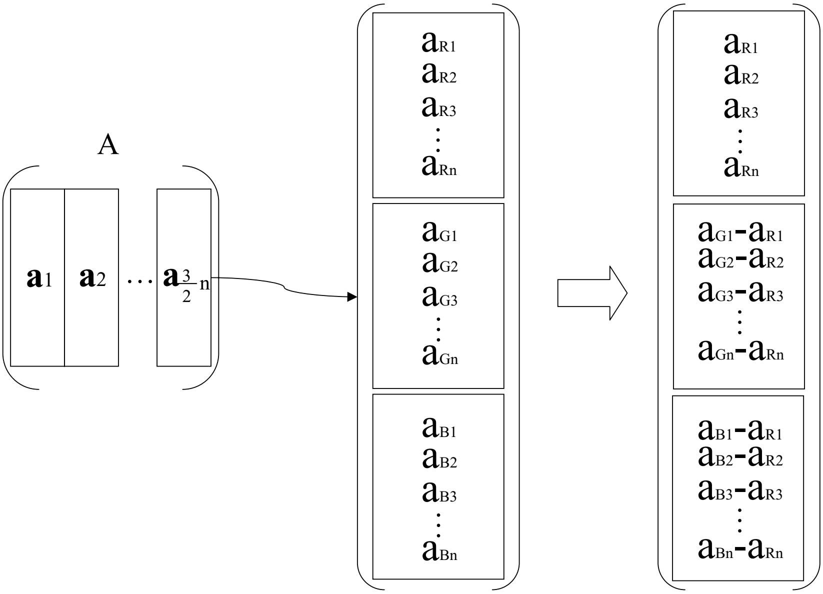


Fig. 2. Computation of bases components' differences

3 File Format with Learned Bases

Figure 3 illustrates the file format which contains headers and compressed information. As can be observed in this figure, the organization of this format is blockwise.

(1) File header:
The file header contains the image size and the offset for each block. But, the file header is free from each block's format so that each block information's independence is maintained.

(2) Information header:
This part is prepared for extra important information which may or may not be related to the image compression. Such information includes the author name and the copyright. Tags like MPEG7 can also be such information.

(3) Average:
The average is used for the image retrieval using color information. This information can be utilized to make thumbnails and progressive expressions.

(4) Bases:
This part contains compressed information of image bases. PCA bases and ICA bases are major targets in this study.

(5) Coefficients:
This part contains compressed information for the superposition of bases.

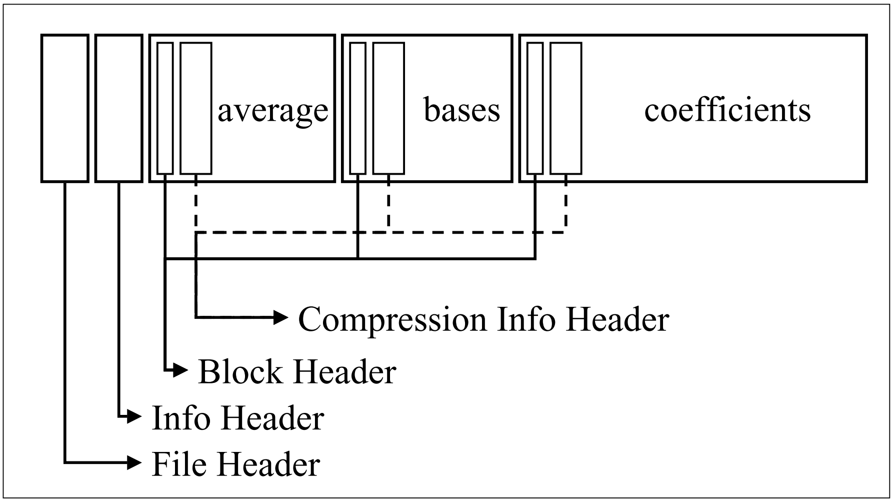


Fig. 3. File format

4 Viewer for Similar Image Retrieval

Figure 4 is a screen shot of the designed viewer Wisvi for the similar-image retrieval. Given a directory name, Wisvi looks for images including all subdirectories. Wisvi is used in the compression performance evaluation and opinion tests for the retrieval.

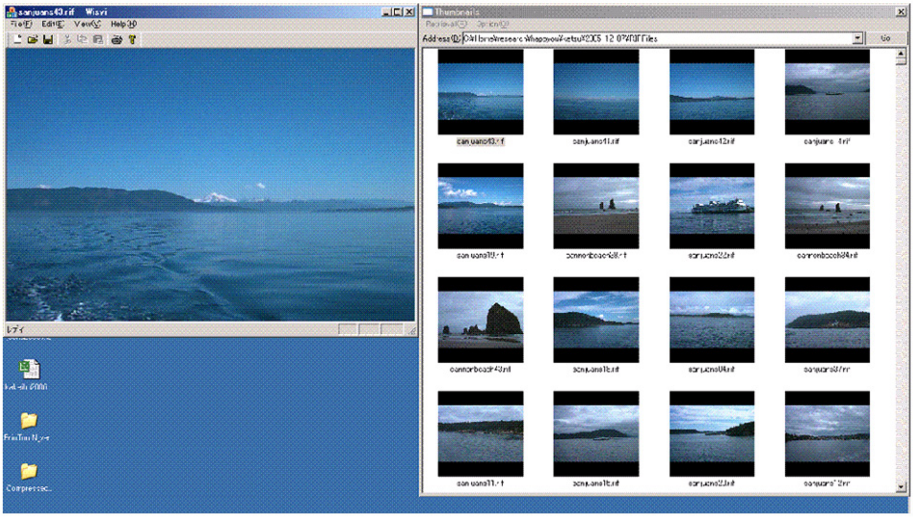


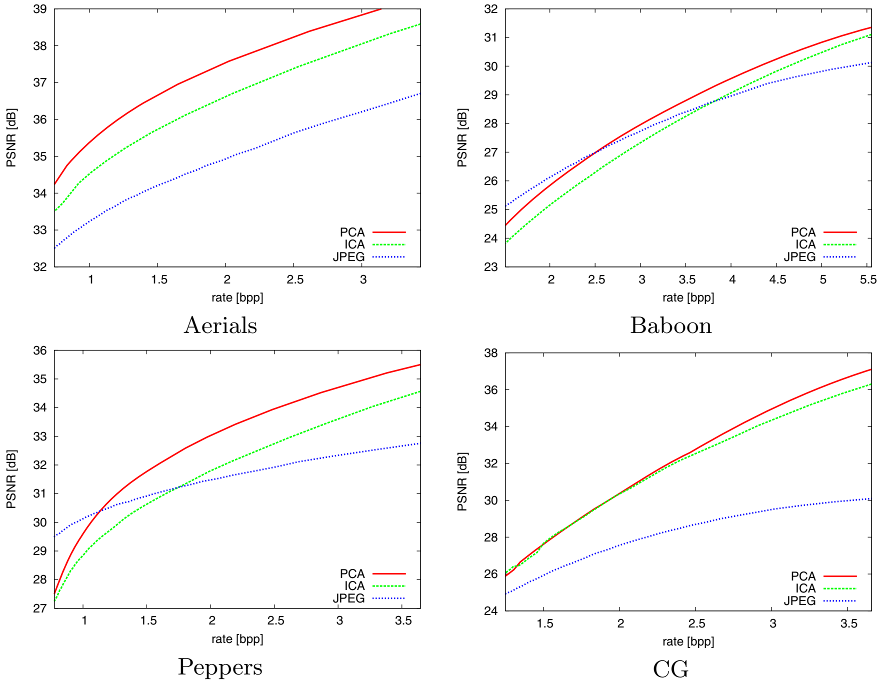
Fig. 4. Wisvi: A similar-image retrieval viewer

The large picture in the upper-left of Figure 4 is the query image. The task is to find similar images from specified directories. Small images on the right side are sorted from the query image to others which are judged similar. It is worthy to note here that the PCA/ICA-based search described in the next section can find similar images with different x-y ratios [2].

5 Preliminary Experiments to Evaluate the Design Principle

Here, we explain why the aforementioned compression method fits to the similar-image retrieval.

(a) Average of colors:
In this system, three averages of {R, G, B} are encoded for the image compression. From an uninitiated intuition, this might look inferior to using a single granular average of the whole color. But, experiments showed that the three component method beats the one average method. This illustration is omitted because of the space limitation.



Aerials

Baboon

Peppers

CG

Fig. 5. Rate-distortion curve

(b) Tight fitting versus universal bases:
Here, "tight fitting bases" stands for the bases from handled images (e.g., the query image). On the other hand, "universal bases" means the bases obtained from a good amount of mixtures of images. Experiments show that the universal bases wins for images with smaller sizes. On the other hand, tightly fitting bases are better for images with larger sizes (illustration is omitted because of the space limitation). Since pixel sizes adopted by recent digital cameras and cellular mobile phones are increasing, this experiment recommends the method of tightly fitting bases.

(c) Compression performance:
Figure 5 illustrates the rate-distortion curves of three compression methods {PCA bases, ICA bases, JPEG} (actually, they are rate-quality curves). These are examples from extensive studies. As can be observed from this figure, the PCA bases outperform the two others. JPEG wins only within low quality ranges. Due to the margin of this win, the joint compression and retrieval of images is made possible.

(d) Similarity measures:
The similarity measure which compare two images is a combination of the color-sensitive part S_{color} and the texture/edge sensitive part S_{bases}.

$$S = \alpha S_{bases} + (1-\alpha) S_{color}, \qquad 0 \leqq \alpha \leqq 1. \tag{17}$$

Here, α is a design parameter set by users (e.g., $\alpha = 0.3$).

The color similarity S_{color} is computed as the average of patch similarity defined by the inner product. The basis similarity S_{bases} is also computed by using the inner product. But, this part needs to consider how to find bases pair to compare for the computational efficiency. Readers are requested refer to [2] for details.

6 Performance of the Similar-Image Retrieval

Figure 6 summarizes the result of the opinion tests by 10 uninitiated users on the Ground Truth Database [5]. This figure compares the retrieval performances by the PCA basis method and the ICA basis method.

The retrieval is judged to be in success if the target to the query image was contained within top $x\,\%$ of all images. This x is called the success line which is the horizontal axis of Figure 6. The vertical axis, the success rate, is measured by showing images one by one to the opinion test subjects. The following summarizes this result on the similar-image retrieval.

(a) Both PCA and ICA methods are judged to be viable.
(b) The ICA basis method outperforms the PCA basis method.
(c) The PCA basis method is faster. This is because the ICA basis computation requires the result of PCA (cf. Section 2).

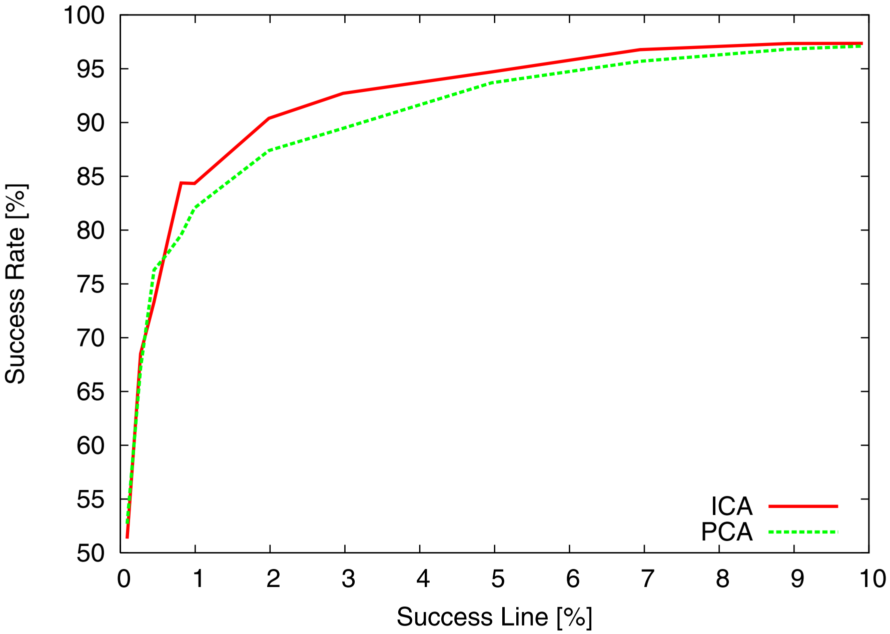


Fig. 6. Retrieval success rates for PCA and ICA methods

7 Conclusion

The retrieval-aware image compression using learned bases was presented. This paper presented the following.

(1) The presented methods for the joint data compression and similar-image retrieval were successful. These methods are based on the learned PCA and ICA image bases.
(2) A basic image format was presented (the rim format; Retrieval-aware IMage format).
(3) The PCA basis method outperforms JPEG for data compression.
(4) Both PCA and ICA bases are successfully retrieval-aware.

This paper leads to the following studies for improvements, which are in progress. Some already show promising results.

(a) The compression experiments in the text used the uniform quantization and the run-length Huffman coding. This part can be improved at the cost of slight increase of computational complexity. The arithmetic coding is one possibility. Therefore, we applied EBCOT (Embedded Block Coding with Optimal Truncation) which is used in JPEG2000. EBCOT comprises the arithmetic coding as the main step. The compression performance was improved. Quantitative results will be given in a separate repots.

(b) Computation speedup for bases using software and/or hardware is desirable.

(c) Comparison with JPEG2000 needs to be studied. JPEG2000 is compression effective. On the compression performance per se, item (a) already gives an answer. On the performance for the retrieval-awareness, additional sophistications are necessary. This is in progress.

(d) Upon this paper is published, a β-version of the tool set for the joint image compression and retrieval will be made downloadable at the URL given in the first page of this paper.

References

1. Flikr: (2005) www.flickr.com
2. Katsumata, N., Matsuyama, Y.: Database retrieval for similar images using ICA and PCA bases. Engineering Applications of Artificial Intelligence **18** (2005) 705–717
3. Hyvärinen, A.: Fast and robust fixed-point algorithm for independent component analysis. IEEE Trans. NN **10** (1999) 626–639
4. Matsuyama, Y., Katsumata, N., Imahara, S.: The alpha-ICA algorithm. Proc. 2000 Int. workshop on ICA and BSS, Espoo, Finland (2000) 297–302
5. Ground Truth Database:
(1999) www.cs.washington.edu/research/imagedatabase/groundtruth

A Suitable Neural Network to Detect Textile Defects

Md. Atiqul Islam[1], Shamim Akhter[1], Tamnun E. Mursalin[1], and M. Ashraful Amin[2]

[1] Department of Computer Science, American International University- Bangladesh, Dhaka, Bangladesh

[2] Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Ave., Kowloon, Hong Kong

islam_md_atiqul@yahoo.com, shamim@aiub.edu, tmursalin@aiub.edu, amin021us@yahoo.com

Abstract. 25% of the total revenue earning is achieved from Textile exports for some countries like Bangladesh. It is thus important to produce defect free high quality garment products. Inspection processes done on fabric industries are mostly manual hence time consuming. To reduce error on identifying fabric defects requires automotive and accurate inspection process. Considering this lacking, this research implements a Textile Defect detector. A multi-layer neural network is determined that best classifies the specific problems. To feed neural network the digital fabric images taken by a digital camera and converts the RGB images are first converted into binary images by restoration process and local threshold techniques, then three different features are determined for the actual input to the neural network, which are the area of the defects, number of the objects in a image and finally the shape factor. The develop system is able to identify two very commonly defects such as Holes and Scratches and other types of minor defects. The developed system is very suitable for Least Developed Countries, identifies the fabric defects within economical cost and produces less error prone inspection system in real time.

Keywords: Textile defects, threshold decision tree, multi-layer neural networks, resilient back propagation, cross validation.

1 Introduction

In the least developed countries like Bangladesh, most defects arising in the production process of a textile material are still detected by human inspection. The work of inspectors is very tedious and time consuming. They have to detect small details that can be located in a wide area that is moving through their visual field. The identification rate is about 70%. In addition, the effectiveness of visual inspection decreases quickly with fatigue. Digital image processing techniques have been increasingly applied to textured samples analysis over the last ten years [1]. Wastage reduction through accurate and early stage detection of defects in fabrics is also an important aspect of quality improvement. Table 1 [2] summarizes the comparison between human visual inspection and automated inspection. Also, it has been observed [3] that price of textile fabric is reduced by 45% to 65% due to defects.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 430 – 438, 2006.

Table 1. Visual inspection versus automated inspection

Inspection Type	Visual	Automated
Fabric Types	100%	70%
Defect Detection Rate	70%	80%+
Reproducibility	50%	90%+
Objective Defect Judgment	50%	100%
Statistics Ability	0%	95%+
Inspection Speed	30 m/min	120 m/min
Response Type	50%	80%
Information Content	50%	90%+
Information Exchange	20%	90%+

In textile sectors, different types of faults are available i.e. hole, scratch, stretch, fly yarn, dirty spot, slub, cracked point, color bleeding etc; if not detected properly these faults can affect the production process massively.

Machine vision automated inspection system for textile defects has been in the research industry for longtime [8], [9]. Recognition of patterns independent of position, size, brightness and orientation in the visual field has been the goal of much recent work. However, there is still a lack of work in machine vision automated system for recognizing textile defects using AI. A neural network pattern recognizer was developed in [10]. Fully connected three multilayer percetron network was used to identify different sizable objects. The input of this network is seven standardized invariant moment and the weights are trained using back propagation. Since the network uses standardized moments as input, neural net similar to this requires lots of iteration to train. The research takes directly input as binary images as a result no preprocessing of image is performed.

Today's automated fabric inspection systems are based on adaptive neural networks. So instead of going through complex programming routines, the users are able to simply scan a short length of good quality fabric to show the inspection system what to expect. This coupled with specialized computer processors that have the computing power of several hundred Pentium chips makes these systems viable [20]. Three state-of-the-art fabric inspection systems are – BarcoVision's Cyclops, Elbit Vision System's I-Tex and Zellweger Uster's Fabriscan. These systems can be criticized on grounds that they all work under structured environments – a feat that is almost non-existent in list developed countries like Bangladesh.

There are some works in [11] based on the optical Fourier transform directly obtained from the fabric with optical devices and a laser beam. Digital image processing techniques have been increasingly applied to textured samples analysis over the last ten years. Several authors have considered defect detection on textile materials. Kang et al. [12], [13] analyzed fabric samples from the images obtained from transmission and reflection of light to determine its interlacing pattern. Wavelets had been applied to fabric analysis by Jasper et al. [14], [15]. Escofet et al. [16], [17] have applied Gabor filters (wavelets) to the automatic segmentation of defects on non-solid fabric images for a wide variety of interlacing patterns. Millán and Escofet [18] introduced Fourier-domain-based angular correlation as a method to recognize similar periodic patterns, even though the defective fabric sample image appeared rotated and scaled. Recognition was achieved when the maximum correlation value of the scaled and rotated power spectra was similar to the autocorrelation of the power spectrum of the pattern fabric sample. If the method above was applied to the spectra presented in Fig.1, the maximum angular correlation value would be considerably lower than the autocorrelation value of the defect free fabric spectrum. Fourier analysis does not provide, in general, enough information to detect and segment local defects.

Electronic textiles (e-textiles) are fabrics with interconnections and electronics woven into them. The electronics consist of both processing and sensing elements, distributed throughout the fabric. Thomas Martin et al. [19] describe the design of a simulation environment for electronic textiles (e-textiles) but having a greater dependence on physical locality of computation. The current status of the simulation environment for e-textiles and present results generated by the environment and associated prototypes for two applications, a large-scale acoustic beam forming fabric for locating vehicles and a pair of pants for classifying and analyzing wearer motions. Gabor filter is a widely feature extraction method, especially in image texture analysis. The selection of optimal filter parameters is usually problematic and unclear. Yimiing et al. [21] analyze the filter design essentials and proposes two different methods to segment the Gabor filtered multi-channel images. The first method integrates Gabor filters with labeling algorithm for edge detection and object segmentation. The second method uses the K-means clustering with simulated annealing for image segmentation of a stack of Gabor filtered multi-channel images. But the classic Gabor expansion is computationally expensive and since it combines all the space and frequency details of the original signal, it is difficult to take advantage of the gigantic amount of numbers. From the literature it is clear that there exists many systems that can detect textile defects but hardly affordable by the small industries of the List Developed countries like Bangladesh.

In this paper we propose a textile defect recognizer that can detect three types of very common faults in textile production, that are hole, scratch, and other fault. An automated textile defect detector based on computer vision methodology and adaptive neural networks is built combining engines of image processing and artificial neural networks in textile industries research arena.

Here the textile defect recognizer is viewed as a real-time control agent that transforms the captured digital image into adjusted resultant output and operates the automated machine (i.e. combination of two leaser beams and production machine),

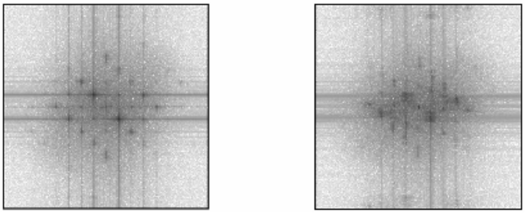


Fig. 1. Power spectrum of the pattern fabric sample (left) and the defective fabric sample (right)

In the proposed system as the recognizer identifies a fault of any type mentioned above, will immediately recognize the type of fault which in return will trigger the laser beams in order to display the upper offset and the lower offset of the faulty portion. The upper offset and the lower offset implies the 2 inches left and 2 inches right offset of faulty portion. This guided triggered area by the laser beans will indicate the faulty portion that needs to be extracted from the roll. After cutting the desired portions of fabric, textile defect recognizer resumes its operation.

2 Mythology and Implementation of the System

Major steps required to implement the proposed system is depicted in Fig. 2. The proposed system can be a competitive model for recognizing textile defects in real world. Base on the research, the proposed system design is separated into two parts. The first part of our research focuses on the processing of the images to prepare to feed into the neural network. The second part is about building a neural network that best performs on the criteria to sort out the textile defects.

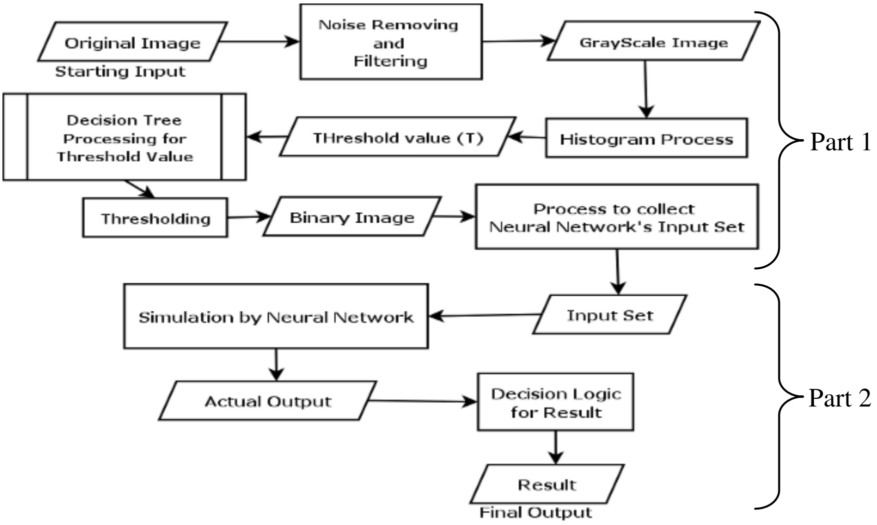

Original Image
Starting Input
Noise Removing and Filtering
GrayScale Image
Decision Tree Processing for Threshold Value
THreshold value (T)
Histogram Process
Part 1
Thresholding
Binary Image
Process to collect Neural Network's Input Set
Simulation by Neural Network
Input Set
Actual Output
Decision Logic for Result
Part 2
Result
Final Output

Fig. 2. Major components of the textile defect detector

2.1 Processing Textile Image for the Neural Network Input

At first the images of the fabric is captured by digital camera in RGB format (top left image in figure and figure) and passes the image through serial port to the computer. Then, noise is removed using standard techniques and an adaptive median filter algorithm has been used as spatial filtering for minimizing time complexity and maximizing performance [4] to converts digital (RGB) images to grayscale images (top middle image in Fig. 3). After restoration local thresholding technique (the process is discussed in next sub-section) is used in order to convert grayscale image into binary image (top right in Fig. 3). Finally, this binary image is used to calculate the following attributes:

1. **The area of the faulty portion:** calculates the total defected area of a image.
2. **Number of objects:** uses image segmentation to calculate the number of labels in an image.
3. **Shape factor:** distinguishes a circular image form a noncircular image. Shape Factor uses the area of a circle to identify the circular portions of the fault.

These three attributes are used as input sets to adapt the neural net through training set in order to recognize expected defects.

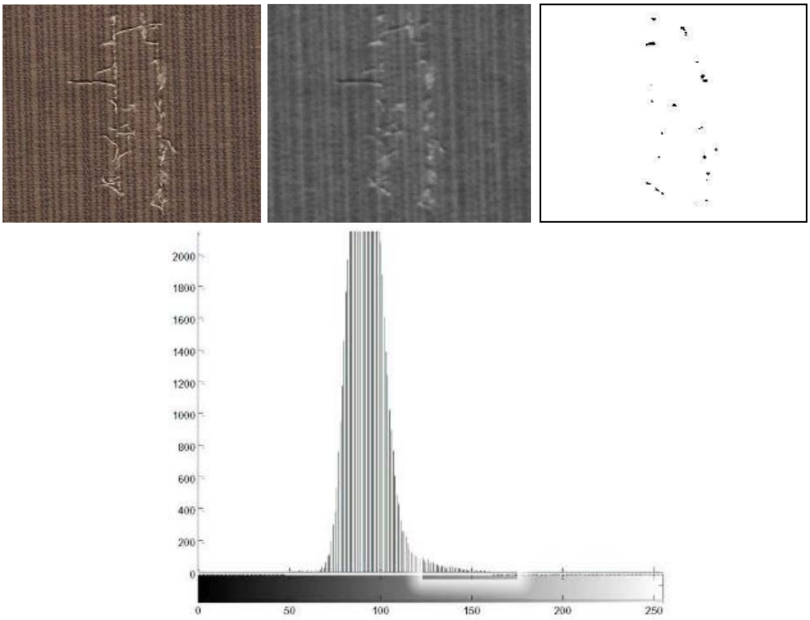


Fig. 3. Original Faulty (Scratch) Fabric (top left), gray (top middle) and binary (top right) representation and histogram (bottom) of the gray

Decision tree for threshold from gray to binary. A decision tree is constructed based on the histogram of the image in hand to convert the gray scale image in a binary representation. As we know from the problem description that there are different types of textile fabrics and also different types of defects in textile industries hence different threshold values to different pattern of faults there is no way to generalize threshold value (T) from one image for all types of fabrics. Notice this phenomenon in histograms illustrated in Fig. 3. (The identified threshold value (T) should be greater then 120 and less than 170) and Fig. 4. (The identified threshold value (T) should be greater then 155 and less then 200). A local threshold was used based on decision tree which was constricted using set of 200 image histograms of fabric data. Illustration of the decision tree is provided in Fig. 5.

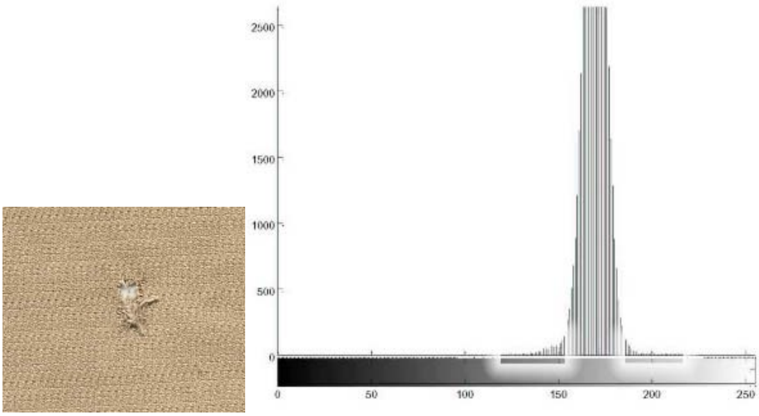


Fig. 4. Original Faulty (Hole) Fabric (left) and the histogram of the gray representation (right)

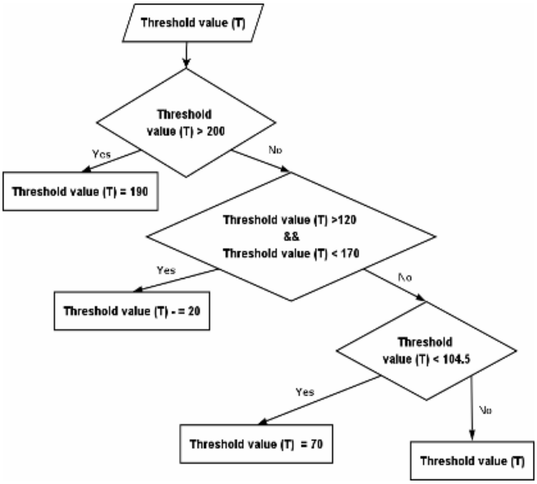

Threshold value (T)
Threshold value (T) > 200
Yes
No
Threshold value (T) = 190
Threshold value (T) >120 && Threshold value (T) < 170
Yes
No
Threshold value (T) - = 20
Threshold value (T) < 104.5
Yes
No
Threshold value (T) = 70
Threshold value (T)

Fig. 5. Decision Tree for Threshold Value (T) to convert from gray to binary

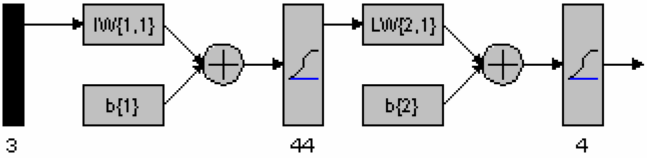


Fig. 6. Design of Feed Forward Back propagation Neural Network

2.2 The Suitable Neural Network

In search of a fully connected multi-layer neural network that will sort out the defected textiles we start with a two layer neural network (Fig. 6). Our neural network contains one hidden of 44 neurons and one output layer of 4 neurons. The neurons in the output layer is delegated as 1st neuron of the output layer is to Hole type fault, 2nd neuron of the output layer is to Scratch type fault, 3rd neuron of the output layer is to Other type of fault and 4th neuron of the output layer is for No fault (not defected fabric). The output range of the each neuron is in the range of [0 ~ 1] as we use log-sigmoid threshold function to calculate the final out put of the neurons. Although during the training we try to reach the following for the target output [{1 0 0 0}, {0 1 0 0}, {0 0 1 0}, {0 0 0 1}] consecutively for Hole type defects, Scratch type defects, Other type defects and No defects, the final output from the output layer is determined using the winner- take-all method.

To determine the number of optimal neurons in the hidden layer was the tricky part, we start with 20 neurons in the hidden layer and test the performance of the neural network on the basis of a fixed test set, and then we increase the number of neurons one by one and till 60, the number of neurons in the hidden layer is chosen based on the best performance. The error curve is illustrated in Fig. 7.

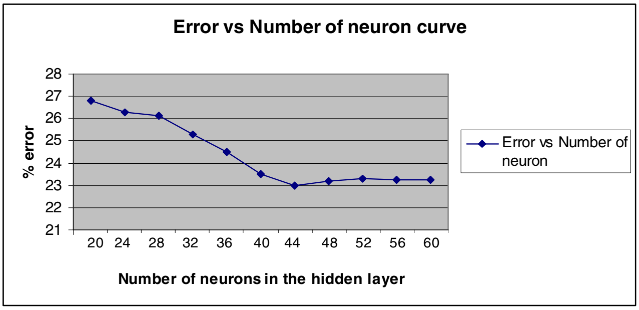


Fig. 7. Performance (in % error) carve on the neuron number in the hidden layer

The parameters used in the neural network can be summarized as:

- Training data set contains 200 images; 50 from each class.
- Test data set contains 20 images; 5 from each class
- The transfer function is Log Sigmoid.
- Performance function used is mean square error
- Widrow-Hoff algorithm is used as learning function [5] with a learning rate of .01.
- To train the network resilient back propagation algorithm [6], [7] is used. Weights and biases are randomly initialized. Initial delta is set to .05 and the maximum value for delta is set to 50, the decay in delta is set to .2.
- Training time or total iteration allowed for the neural networks to train is set to infinity as we know it is a conversable problem. And we have the next parameter to work as stopping criterion
- Disparity or maximum error in the actual output and network output is set to 10^{-5}.

3 Results and Discussions

The performance of the textile recognizer is determined based on the cross validation method. The average result is provided in Fig. 8. Here notice that the recognizer can successfully identifying Hole type faults with 72% accuracy, 65% of Scratch type faults, 86% of the Other type faults and 83% No faults accurately. The average performance of the system determining the defects in textile industry is 74.33% and the overall all performance of the system is 76.5%.

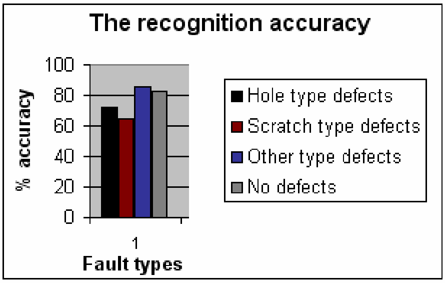


Fig. 8. The bar chart for the performance accuracy of the system

4 Conclusion

All textile industries aim to produce competitive fabrics. The competition enhancement depends mainly on productivity and quality of the fabrics produced by each industry. In the textile sector, there have been an enlarge amount of losses due to faulty fabrics. Here we have demonstrated that Textile Defect Recognition System is capable of detecting fabrics' defects with more accuracy and efficiency. In the research arena, our system tried to use the local threshold technique without the decision tree process.

The system performs quite well except the problem of false negative classification, where it fails to classify the good fabric as good and marks it as faulty fabric; the future versions of the system will try to notice this problem more precisely.

References

1. M. Ralló, M. S. Millán, J. Escofet, "Wavelet based techniques for textile inspection", Opt. Eng. 26(2), 838-844 (2003)
2. R. Meier, "Uster Fabriscan, The Intelligent Fabric Inspection," [Online document], cited 20 Apr. 2005], Available HTTP: http://www.kotonline.com/english_pages/ana_basliklar/uster.asp
3. R. Stojanovic, P. Mitropulos, C. Koulamas, Y. A. Karayiannis, S. Koubias, and G. Papadopoulos, "Real-time Vision based System for Textile Fabric Inspection", Real-Time Imaging, vol. 7, no. 6, 2001, pp. 507–518.

4. R. C. Gonzalez, R. E. Woods, S. L. Eddins, “Digital Image Processing using MATLAB”, ISBN 81-297-0515-X, 2005, pp. 76-104,142-166,404-407
5. M. T. Hagan, H. B. Demuth, M. Beale, “Neural Network Design”, ISBN 981-240-376-0, 2002, part 2.5, 10.8
6. Riedmiller, M., and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm", Proceedings of the IEEE International Conference on Neural Networks, 1993.
7. Neural Network Toolbox, “MATLAB –The Language of Technical Conputing”, [CD Document], Version 7.0.0.19920(R14), 2004
8. B. G. Batchelor and P. F. Whelan, “Selected Papers on Industrial Machine Vision Systems,” SPIE Milestone Series, 1994.
9. T. S. Newman and A. K. Jain, “A Survey of Automated Visual Inspection,” Computer Vision and Image Understanding, vol. 61, 1995, pp. 231–262.
10. H Zhang, J. Guan and G. C. Sun, “Artificial Neural Network-Based Image Pattern Recognition”, ACM 30th Annual Southeast Conference, 1992
11. Ciamberlini C., Francini F., Longobardi G., Sansoni P., Tiribilli, B. "Defect detection in textured materials by optical filtering with structured detectors and selfadaptable masks", Opt. Eng. 35(3), 838-844 (1996)
12. Kang T.J. et al. “Automatic Recognition of Fabric Weave Patterns by Digital Image Analysis”, Textile Res. J. 69(2), 77-83 (1999)
13. Kang T.J. et al. “Automatic Structure Analysis and Objective Evaluation of Woven Fabric Using Image Analysis”, Textile Res. J. 71(3), 261-270 (2001)
14. Jasper W.J., Garnier S.J., Potlapalli H., "Texture characterization and defect detection using adaptive wavelets", Opt. Eng. 35(11), 3140-3149 (1996)
15. Jasper W.J., Potlapalli H., "Image analysis of mispicks in woven fabric", Text. Res.J. 65(1), 683-692 (1995)
16. Escofet J., Navarro R., Millán M.S., Pladellorens J., "Detection of local defects in textile webs using Gabor filters", in "Vision Systems: New Image Processing Techniques" Ph. Réfrégier, ed. Proceedings SPIE vol. 2785, 163-170 (1996)
17. Escofet J., Navarro R., Millán M.S., Pladellorens J., "Detection of local defects in textile webs using Gabor filters", Opt. Eng. 37(8) 2297-2307 (1998)
18. Millán M.S., Escofet J., “Fourier domain based angular correlation for quasiperiodic pattern recognition. Applications to web inspection”, Appl. Opt. 35(31), 6253-6260 (1996)
19. T. Martin, M. Jones, J. Edmison, T. Sheikh and Z. Nakad,”Modeling and Simulating Electronic Textile Applications”, LCTES, USA, 2004
20. A. Dockery, “Automatic Fabric Inspection: Assessing the Current State of the Art,” [Online document], 2001, [cited 29 Apr. 2005], Available HTTP:
21. Y. Ji, K. H. Chang and CC. Hung, “Efficient Edge Detection and Object Segmentation Using Gabor Filters”, ACMSE, USA, 20041.

MPEG Video Traffic Modeling and Classification Using Fuzzy C-Means Algorithm with Divergence-Based Kernel

Chung Nguyen Tran and Dong-Chul Park

ICRL, Dept. of Information Engineering, Myong Ji University, Korea
{tnchung, parkd}mju.ac.kr

Abstract. A modeling and classification model for MPEG video traffic data using a Fuzzy C-Means algorithm with a Divergence-based Kernel (FCMDK) for clustering GPDF data is proposed in this paper. The FCMDK is based on the Fuzzy C-Means clustering algorithm and thus exploits advantageous features of fuzzy clustering techniques. To further improve classification accuracies and deal with nonlinear data, the input data is projected into a feature space of a higher dimensionality. Consequently, nonlinear problems existing in the input space can be solved linearly in the feature space. The divergence-based kernel method adopted in the FCMDK employs a divergence measure between two probability distributions for its similarity measure. By adopting the divergence-based kernel method for probability data, the FCMDK can not only utilize advantageous features of the kernel method but also exploit the statistical nature of the input data. Experiments and results on several MPEG video traffic data sets demonstrate that the classification model employing the FCMDK for clustering GPDF data can archive improvements of 28.19% and 34.60% in terms of False Alarm Rate (FAR) over the models using the conventional k-means and SOM algorithms, respectively.

1 Introduction

Content-based retrieval of video data has attracted a great attention in recent years. Many video applications such as video on demand, video databases, and video teleconferencing can benefit from retrieval of the video data based on their content. However, with the rapid increase in various multimedia services, numerous video databases are available through the internet. Organizing these huge video databases into libraries and providing effective indexing require an efficient modeling and classification model.

Recently, various video data classification models have been proposed [1,2,3,4]. Most of classification models are based on pattern recognition approaches which often use a Gaussian Mixture Model (GMM) for modeling video traffic data and a Bayesian classifier [4]. In order to obtain mixture components , also called Gaussian Probability Density Function (GPDF) data, in GMMs, clustering algorithms are often employed. For clustering GPDF data, conventional Self Organizing Map (SOM) [5] and k-means [6] algorithms have been most widely used in

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 439–447, 2006.

practice because of their simplicity. Later, the Fuzzy C-Means (FCM) clustering algorithm is proposed as an improvement of the k-means and the SOM [7,8]. The FCM has been successfully applied in clustering the probabilistic distribution of the log-value of the frame size in the MPEG video classification model proposed by Liang and Mendel [4]. However, these algorithms were designed with the Euclidean distance. This implies that most of video classification models using these clustering algorithms used only mean values of GPDF data for clustering while leaving out covariance information of GPDF data. To exploit entire information in data including the mean value and covariance information, Park and Kwon proposed a divergence-based centroid neural network (DCNN) algorithm for clustering GPDF data [9]. The DCNN has been successfully applied to the clustering GPDF data for Hidden Markov Model (HMM) in speech applications.

In this paper, a MPEG video traffic classification model using a Fuzzy C-Means Algorithm with a Divergence-based Kernel (FCMDK) is proposed. The proposed classification model is designed for the classification of compressed video data without going through the decompressing procedure. The FCMDK adopted in the proposed classification model is used for clustering the GPDF data. The FCMDK is based on the FCM algorithm and thus utilizes advantageous features of fuzzy clustering techniques. Before clustering, the input data is projected to a feature space using a kernel method. The kernel method adopted in the FCMDK is used to transform the input data from a low dimensional space to a feature space of a higher dimensionality [10,11]. Consequently, nonlinear problems associated with the input space can be solved linearly in the feature space according to the well-known Mercer theorem [12]. Furthermore, the statistical nature of the data is utilized by using both the mean value and covariance information in GPDF data. For clustering of probability data, a divergence-based kernel using a divergence measure as its measure distance between two probability distributions is employed.

The remainder of this paper is organized as follows. Section 2 summarizes the Fuzzy C-Means and the Kernel-based Fuzzy C-Means algorithms. Section 3 introduces the Fuzzy C-Means algorithm with Divergence-based Kernel. Section 4 presents experiments and results on several MPEG video data sets including comparisons with other conventional algorithms. Conclusions are presented in Section 5.

2 Kernel-Based Fuzzy C-Means Algorithm

2.1 Fuzzy C-Means Algorithm

The FCM algorithm has successfully been applied to a wide variety of clustering problems. The FCM algorithm attempts to partition a finite collection of elements $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ into a collection of C fuzzy clusters. Bezdek first generalized the *fuzzy ISODATA* by defining a family of objective functions $J_m, 1 < m < \infty$, and established a convergence theorem for that family of objective functions [7,8]. For the FCM, the objective function is defined as :

$$J_m(U, \boldsymbol{v}) = \sum_{i=1}^{C} \sum_{k=1}^{N} \mu_{ik}^{m} \|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2 \tag{1}$$

where $\|.\|^2$ denotes Euclidean distance measure, $\boldsymbol{x}_k$ and $\boldsymbol{v}_i$ is the input data, k, and cluster prototype, i, respectively. μ_{ki} is the membership grade of the input data $\boldsymbol{x}_k$ to the cluster $\boldsymbol{v}_i$, and m is the weighting exponent, $m \in \{1, \cdots, \infty\}$, while N and C are the number of input data and clusters, respectively.

The FCM objective function is minimized when high membership grades are assigned to objects which are close to their centroid and low membership grades are assigned when objects are far from their centroid [8].

By using the Lagrange multiplier to minimize the objective function, the center prototypes and membership grades can be updated as follows:

$$\mu_{ik} = \frac{1}{\sum\limits_{j=1}^{C} \frac{\|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2}{\|\boldsymbol{x}_k - \boldsymbol{v}_j\|^2}} \tag{2}$$

$$\boldsymbol{v}_i = \frac{\sum\limits_{k=1}^{N} \mu_{ik}^{m} \boldsymbol{x}_k}{\sum\limits_{k=1}^{N} \mu_{ik}^{m}} \tag{3}$$

The FCM finds the optimal values of group centers iteratively by applying Eq. (2) and Eq. (3) in an alternating fashion.

2.2 Kernel-Based Fuzzy C-Means Algorithm

Though the FCM has been applied to numerous clustering problems [13], it still suffers from poor performance when boundaries among clusters in the input data are nonlinear. One alternative approach is to transform the input data into a feature space of a higher dimensionality using a nonlinear mapping function so that nonlinear problems in the input space can be linearly treated in the feature space according to the well-known Mercer theorem [12,11]. One of the most popular data transformation methods adopted in recent studies is the kernel method [10]. One of the advantageous features of the kernel method is that input data can be implicitly transformed into the feature space without knowledge of the mapping function. Further, the dot product in the feature space can be calculated using a kernel function.

With the incorporation of the kernel method, the objective function in the feature space using the mapping function Φ can be rewritten as follow:

$$F_m = \sum_{i=1}^{C} \sum_{k=1}^{N} \mu_{ik}^{m} \|\Phi(\boldsymbol{x}_k) - \Phi(\boldsymbol{v}_i)\| \tag{4}$$

Through kernel substitution, the objective function can be rewritten as:

$$F_m = 2 \sum_{i=1}^{C} \sum_{k=1}^{N} \mu_{ik}^{m} (1 - K(\boldsymbol{x}_i, \boldsymbol{v}_k)) \tag{5}$$

where $K(\boldsymbol{x}, \boldsymbol{y})$ is a kernel function used for calculating the dot product of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ in the feature space. To calculate the kernel between two vectors, the Gaussian kernel function is widely used:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{\sigma^2}\right) \tag{6}$$

By using the Lagrange multiplier to minimize the objective function, the cluster prototypes can be updated as follow:

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{N} \mu_{ik}^m K(\boldsymbol{x}_k, \boldsymbol{v}_i)\boldsymbol{x}_k}{\sum_{k=1}^{N} \mu_{ik}^m K(\boldsymbol{x}_k, \boldsymbol{v}_i)} \tag{7}$$

And the membership grades can be updated as follow:

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{C} \left(\frac{1-K(\boldsymbol{x}_k, \boldsymbol{v}_i)}{1-K(\boldsymbol{x}_k, \boldsymbol{v}_j)}\right)^{\frac{1}{m-1}}} \tag{8}$$

3 Fuzzy C-Means Algorithm with Divergence-Based Kernel

Since conventional kernel-based clustering algorithms were designed for deterministic data, they cannot be used for clustering probability data. In this paper, we propose a Fuzzy C-Means algorithm with a Divergence-based Kernel (FCMDK) in which a divergence distance is employed to measure the distance between two probability distributions. The proposed FCMDK incorporates the FCM for clustering data and the divergence-based kernel method for data transformation.

For GPDF data, each cluster prototype is not represented by a deterministic vector in the input space but is represented by a GPDF with a mean vector and covariance matrix. In order to calculate the kernel between two GPDF data, a divergence-based kernel is employed. The divergence-based kernel is an extension of the standard Gaussian kernel. While the Gaussian kernel is the negative exponent of the weighted Euclidean distance between two deterministic vectors as shown in Eq. 6, the divergence-based kernel is the negative exponent of the weighted divergence measure between two GPDF data. The divergence-based kernel function between two GPDF data is defined as follows:

$$DK(g_{\boldsymbol{x}}, g_{\boldsymbol{y}}) = \exp\left(-\alpha D(g_{\boldsymbol{x}}, g_{\boldsymbol{y}}) + \beta\right) \tag{9}$$

where $DK(g_{\boldsymbol{x}}, g_{\boldsymbol{y}})$ is the divergence distance between two Gaussian distributions, $g_{\boldsymbol{x}}$ and $g_{\boldsymbol{y}}$. α and β are the constants which depend on the data. After evaluating several divergence distance measures, the popular Bhattacharyya distance measure is employed. The similarity measure between two distributions using the Bhattacharyya distance measure is defined as follows:

$$D(G_i, G_j) = \frac{1}{8}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \left[\frac{\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j}{2}\right]^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + \frac{1}{2} \ln \frac{\left|\frac{\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j}{2}\right|}{\sqrt{|\boldsymbol{\Sigma}_i|\,|\boldsymbol{\Sigma}_j|}} \tag{10}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ denote the mean vector and covariance matrix of a Gaussian distribution G_i, respectively. T denotes the transpose matrix.

Similar to the cluster prototypes and membership grades in the kernel-based FCM, the cluster prototypes and membership grades in the FCMDK can be updated using a Lagrange multiplier to minimize its objective function. However, each cluster prototype representing a cluster in the FCMDK is a probability distribution with a mean vector and a covariance matrix. Therefore, cluster prototypes in each iteration are updated by modifying their mean vector and covariance matrix as follows:

$$m_{\boldsymbol{v}_i} = \frac{\sum\limits_{k=1}^{N} \mu_{ik}^m DK(\boldsymbol{x}_k, \boldsymbol{v}_i) m_{\boldsymbol{x}_k}}{\sum\limits_{k=1}^{N} \mu_{ik}^m DK(\boldsymbol{x}_k, \boldsymbol{v}_i)} \tag{11}$$

$$\Sigma_{\boldsymbol{v}_i} = \frac{\sum\limits_{k=1}^{N} \mu_{ik}^m DK(\boldsymbol{x}_k, \boldsymbol{v}_i) \Sigma_{\boldsymbol{x}_k}}{\sum\limits_{k=1}^{N} \mu_{ik}^m DK(\boldsymbol{x}_k, \boldsymbol{v}_i)} \tag{12}$$

where $m_{\boldsymbol{v}_i}$ and $m_{\boldsymbol{x}_k}$ are the mean of the cluster prototype $\boldsymbol{v}_i$ and the vector in input $\boldsymbol{x}_k$, respectively. $\Sigma_{\boldsymbol{v}_i}$ and $\Sigma_{\boldsymbol{x}_k}$ are the covariance of the cluster prototype $\boldsymbol{v}_i$ and the vector in input $\boldsymbol{x}_k$, respectively. $DK(\boldsymbol{x}_k, \boldsymbol{v}_j)$ is the divergence-based kernel function between two Gaussian distributions $\boldsymbol{x}_k$ and $\boldsymbol{v}_j$.

The membership grades are similar to those in the KFCM and can be updated as follows:

$$\mu_{ik} = \frac{1}{\sum\limits_{j=1}^{c} \left(\frac{1 - DK(\boldsymbol{x}_k, \boldsymbol{v}_i)}{1 - DK(\boldsymbol{x}_k, \boldsymbol{v}_j)}\right)^{\frac{1}{m-1}}} \tag{13}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{v}_i$ are the probability distribution input vector and probability distribution cluster prototype, respectively. $DK(\boldsymbol{x}_k, \boldsymbol{v}_j)$ is the divergence-based kernel function between two Gaussian distributions, $\boldsymbol{x}_k$ and $\boldsymbol{v}_j$.

With the incorporation of the divergence-based kernel method and the FCM, the proposed FCMDK can be used for clustering GPDF data while utilizing the advantageous features of the fuzzy clustering techniques and the kernel method. Thus, it provides an efficient clustering algorithm for GPDF data.

4 Experiments and Results

To demonstrate the performance of MPEG video traffic classification model using the FCMDK, several MPEG video traces were used for experiments. These MPEG video traces were coded with the MPEG-1 standard according to the Moving Picture Expert Group. Table 1 shows the list of video traces used in our experiments. These data are provided by the University of Wuerzburg, Wuerzburg, Germany and are available at the following website:

http://www3.informatik.uni-wuerzburg.de/MPEG/

Table 1 consists of 5 *"movie"* traces and 5 *"sports"* traces. Each trace consists of 40,000 frames which result in 3,333 GOPs. Each GOP can be represented by the sequence *IBBPBBPBBPBB* with 12 frames for each GOP. More details on these video traces can be found in [14].

Table 1. MPEG-1 Video used for experiments

MOVIE	SPORTS
"Jurassic Park"	*"ATP Tennis Final"*
"The Silence of the Lambs"	*"Formula 1 Race: GP Hockenheim 1994"*
"Star Wars"	*"Super Bowl Final 1995: SanDiego-San Francisco"*
"Terminator 2"	*"Two 1993 Soccer World Cup Matches"*
"A 1994 Movie Preview"	*"Two 1993 Soccer World Cup Matches"*

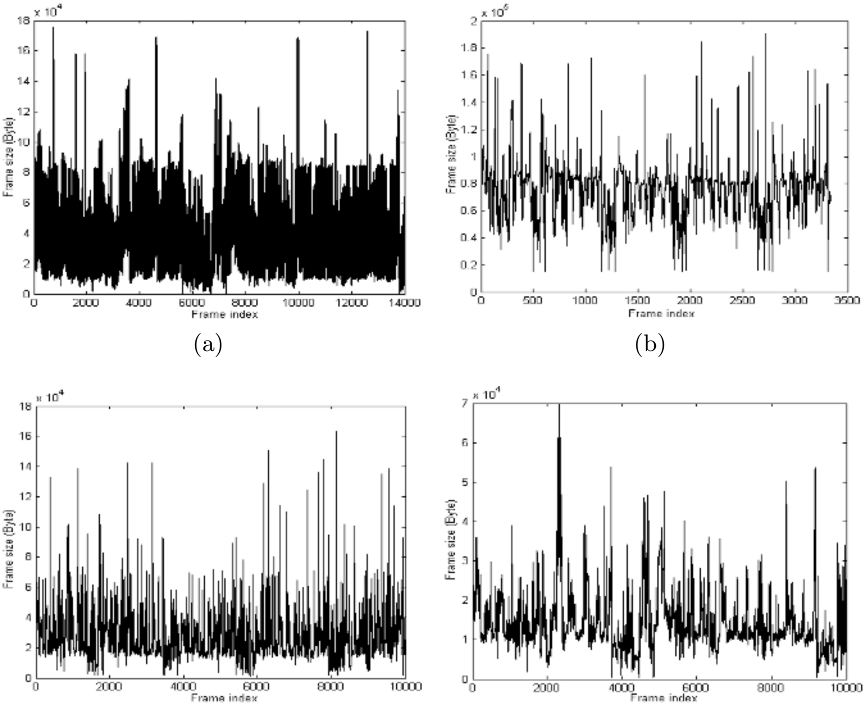
From video traces in Table 1, we used the first 24,000 frames, resulting in 2,000 GOPs from each trace, for training and the remaining frames from each trace for testing. Fig. 1 shows an example of MPEG-1 data with I-, P-, and B-frame from the video data *"Two 1993 soccer World Cup matches"*.

The proposed classification model using the FCMDK is based on a Gaussian Mixture Model (GMM) and a Bayesian classifier. In order to model and classify the MPEG video data, we consider the MPEG data as Gaussian Probability Density Function (GPDF) data [14]. The classification process of proposed classification model can be divided into two steps: the modeling step and the classification step. In the modeling step, mixture components of GMMs are obtained using the FCMDK algorithm. Then, in the classification step, a Bayesian classifier is employed to decide the genre, *"movie"* or *"sports"*, to which a video sequence belongs. The genre decision procedure can be summarized by the following equations:

$$Genre(x) = \arg\max_i P(x|v_i) \tag{14}$$

$$P(x|v_i) = \sum_{i=1}^{M} c_i \aleph(x, \mu_i, \Sigma_i) \tag{15}$$

$$\aleph(x, \mu_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} e^{-0.5(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)} \tag{16}$$



(a) (b) (c) (d)

Fig. 1. Example of MPEG data: (a) whole data (b) I-frame (c) P-frame (d) B-frame

where M is the number of code vectors, c_i is weight of the code vectors, d is the number of dimensions of feature vectors (d = 12), and m_i and Σ_i are the mean and covariance matrix of the i-th group of the genre's distribution, respectively.

In order to evaluate the performance of the proposed classification model, the classification performance is measured by the False Alarm Rate (FAR) which is calculated by the following equation:

$$\text{FAR}(\%) = \frac{\text{Number of misclassification GOPs}}{\text{Total number of GOPs}} \times 100 \qquad (17)$$

One of the most important parameters that has to be selected in most clustering algorithms is the number of clusters in the data. Most clustering algorithms partition data into a specified number of clusters, regardless of whether the clusters are meaningful. In our experiments, the number of code vectors is varied from 3 to 8 in order to determine a sufficient number of code vectors to represent the number of mixture components in the GMMs. Fig. 2 shows the classification performance in terms of FAR of classification models using the SOM, the k-means, and the FCMDK. As can be seen from Fig. 2, the FARs of all classification models are decreased significantly when the number of code vectors is increased from 3 to 5 while they tend to saturate when the number of code

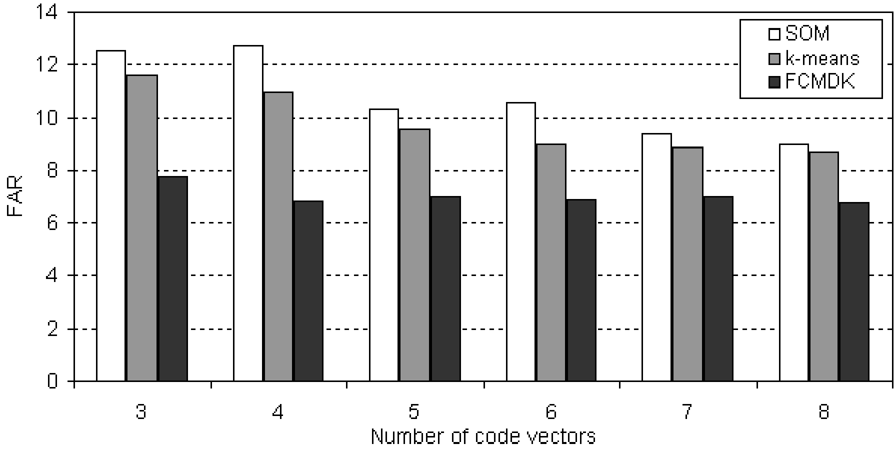


Fig. 2. Overall classification accuracies using different algorithms

Table 2. Average FAR (%) of different classification models

	Overall FAR(%)
SOM	**10.748**
k-means	**9.789**
FCMDK	**7.029**

vectors is greater than 5. This implies that using 5 code vectors for representing the number of mixture components is sufficient.

Table 2 summarizes the classification performance in terms of FAR for different models using the SOM, the k-means, and the proposed FCMDK. As can be seen from Table 2, the classification model using the proposed FCMDK outperforms the models using the SOM and k-means. Improvements in terms of FAR of 28.19% and 34.60% are achieved over the k-means and the SOM algorithms, respectively. These results imply that the covariance information plays an important role in modeling and classification of MPEG video traffic data. By using divergence-based kernel, the FCMDK can utilize the covariance information of the GPDF data for clustering. Thus, it can be used as an efficient tool for clustering GPDF data in GMMs.

5 Conclusion

A new approach for modeling and classification of MPEG video traffic data using a Fuzzy C-Means (FCMDK) algorithm with a Divergence-based Kernel is proposed in this paper. The proposed classification model is based on a Gaussian Mixture Model (GMM) and a Bayesian classifier. The FCMDK adopted in the proposed classification model is employed for clustering of the GPDF data in GMMs. The proposed classification model using the FCMDK for clustering of

GPDFs is applied to a modeling and classification problem of MPEG video traffic data. Our experiments and results for several MPEG video traffic data sets show that respective improvements of 28.19% and 34.60% in terms of FAR are archived over the conventional k-means and the SOM algorithms, respectively. The proposed MPEG video traffic classification model provide an efficient tool for organizing and retrieval of video databases.

References

1. Dawood, A.M., Ghanbari,M.: MPEG Video Modeling Based on Scene Description. IEEE Int. Conf. Image Processing, Chicago, IL., Vol. 2 (1998) 351-355.
2. Manzoni,P.,Cremonesi,P.,Serazzi,G.: Workload models of VBR video traffic and their use in resource allocation policies. IEEE Trans. on Networking 7 (1999) 387-397
3. Krunz, M., Sass, R., Hughes, H.: Statistical Characteristics and Multiplexing of MPEG streams. Proc. IEEE Int. Conf. Comput. Commun., INFOCOM'95, Boston, MA, Vol. 2 (1995) 445-462
4. Liang, Q., Mendel, J.M.: MPEG VBR Video Traffic Modeling and Classification Using Fuzzy Technique. IEEE Trans. on Fuzzy Systems 9 (2001) 183-193
5. Kohonen,T.: The Self-Organizing Map. Proc. IEEE, Vol. 78 (1990) 1464-1480
6. Hartigan, J.: Clustering Algorithms. New York, Wiley (1975)
7. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. New York(Plenum), (1981)
8. Bezdek, J.C.: A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms. IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI-2(1) (1980) 1-8
9. Park, D.C., Kwon, O.H.: Centroid Neural Network with the Divergence Measure for GPDF Data Clustering. IEEE Trans. on Neural Networks (in review).
10. Muller, K.R., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B.: An Introduction to Kernel-Based Learning Algorithms. IEEE Transactions on Neural Networks 12(2) (2001) 181-201
11. Girolami,M.: Mercer Kernel-Based Clustering in Feature Space. IEEE Trans. on Neual Networks 13(3) (2002) 780-784.
12. Cover, T.M.: Geomeasureal and Statistical Properties of Systems of Linear Inequalities in Pattern Recognition. Electron. Computing, Vol. EC-14 (1965) 326-334.
13. Chen, S., Zhang, D.: Robust Image Segmentation using FCM with Spatial Constraints Based on New Kernel-Induced Distance Measure. IEEE Trans. on Systems, Man and Cybernetics 34(4) 2004 1907-1916
14. Rose, O.: Satistical Properties of MPEG Video Traffic and Their Impact on Traffic Modeling in ATM systems. Univ. Wurzburg, Inst. Comput. Sci., Rep., 101 (1995)

A Novel Sports Video Logo Detector Based on Motion Analysis

Hongliang Bai[1], Wei Hu[2], Tao Wang[2], Xiaofeng Tong[2],
Changping Liu[3], and Yimin Zhang[2]

[1] Graduate School, Chinese Academy of Sciences, Automation of Institute
hongliang.bai@ia.ac.cn
[2] Intel China Research Center
{wei.hu, tao.wang, xiaofeng.tong, yimin.zhang}@intel.com
[3] Chinese Academy of Sciences, Automation of Institute
lcp@hanwang.com.cn

Abstract. Replays are key cues for events detection in sport videos since they are the immediate consequence of highlights or important events happened in sports. In many sports videos, replays are usually sandwiched with two identical logo transitions, prompt the beginning and end of a replay. A logo transition is a kind of special digital video effects, usually contains 12-35 consecutive frames, describe a flying or variable object. In this paper, a novel automatic logo detection approach is proposed. It contains two main stages: a logo transition template is automatically learned by dynamic programming and unsupervised clustering, a key frame is also extracted; then the extracted key frame and the learned logo template are used jointly to detect logos in sports videos. The optical flow features are used to depict the motion characteristics of the logo transitions. Experiments on different types of sports videos show that the proposed approach can reliably detect logos in sports videos efficiently.

1 Introduction

Replay is reliable indicator of sports highlight due to the incorporation of expert's judgement [1,2]. In sports videos, a replay can contain slow motion or non-slow motion or both. In the literature, some works were proposed to detect the slow motion replays by the observation that the slow motion replay sequences have the different motion model with the normal sequences [3,4,5]. These approaches have difficulties in slow motion replays generated by high speed cameras, and can not be applied to replays without slow motion. However many sports videos have replays sandwiched by two identical logo transitions. Thus a replay can be located by detecting the logos around of it. The problem of replay detection is then converted to the logo detection. The replays can be located by pairing the detected logos.

It is well known that usually a replay is sandwiched with a special transition at the beginning and end of it, in which a highlighted logo comes in and out quickly. We call this transition as "logo-transition". Some works have been done

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 448–457, 2006.

to detect the logos in sports videos. Pan et al. [6] proposed a replay detection method based on detection of the logos. It first detects two replay segments, then searches two most similar frames that precede the two detected replay segments. And consider the most similar frames are candidate logos. Finally a verification procedure is employed. The problem of this approach is that it needs find confident replay segments first. Duan et al. [7] proposed a logo detection approach based on mean shift which is a kind of motion features. In [8], Tong et al. proposed an approach based on the difference between two consecutive frames. It assumed that the logo was highlighted and located at the center part of a frame.

All above approaches try to find a single key frame which is considered as a representation of a logo. In fact, a single frame is not sufficient to represent a logo, since a logo transition contains a continues movement of a "logo object", and each occurrence of a logo transition in a same sport video are different. A logo transition is generated by superposing a foreground logo object onto the complex variational background. Instead of a single frame, a sequence of logo frames (a logo template) that characterize the whole logo transition is considered here. The key frame is also used to filter out those non-logo positions to speedup the whole detection procedure.

Also a logo transition in a sports video has some additional characteristics that can help to identify:

(1)Occur at the start and end of a replay.

(2)Repeat tens of times in the same video.

(3)During the whole game, the logo object will keep in same.

(4)The logo object usually runs faster than the background, and is different in color, brightness etc. with non-logo frames.

(5)The duration of a logo transition is usually less than 1 second.

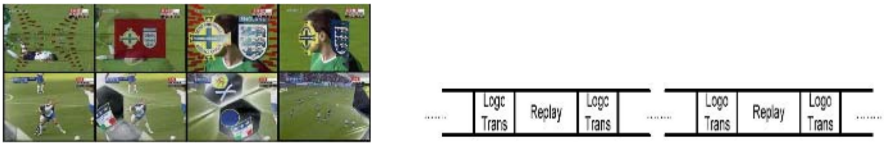
Based on above observations, a robust and generic logo template detector is proposed. The paper is organized as following. Section 2 will introduce the whole system framework. The logo template extraction algorithm will be proposed in section 3. In section 4, the experiment results are shown. Conclusions are given in section 5.

2 System Overview

Logo transitions will occur in many broadcasting sports videos, such as the World Cup, UEFA Champions League, Olympics Game. Figure 1(a) shows two logo transitions. In the sports video, logo transitions pairwise occur, shown in Figure 1(b), and the replay is sandwiched by these two logos.

The whole system of logo detection contains two main stages: training stage and detection stage, shown in Figure 2.

In the training stage, first the video is parsed by a shot boundary detection tool, some of the logo transitions can be labeled as Gradual Transition (GT). Note that the exactly boundary of each GT is not required in this system. Motion features are calculated for each frame in these GT sequences, then dynamic

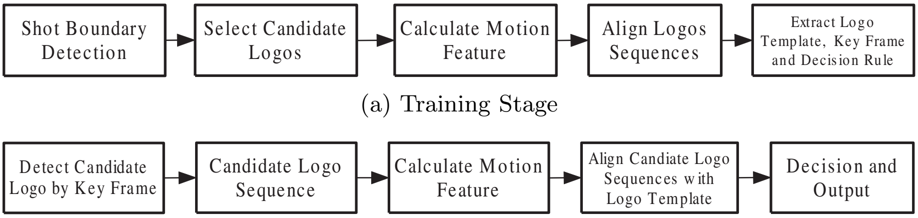


(a)Two Logos Transitions (b) Logo Transition and Replay

Fig. 1. Examples of Logo Transition

programming is performed to align each pair of the GT sequences. A gaussian mixture model (GMM) is applied to determine if each GT is a candidate logo. A logo template is then selected from those candidate logo sequences. At the same time, a key frame in the logo template are extracted, and the decision rule that to judge if a sequence is a real logo is also determined.

In the detection stage, the whole video is scanned by the extracted key frame, and all the candidate logo positions are determined. Each this candidate sequence then be verified by aligning with the logo template that generated in the training stage. Those sequences that accord with the decision rule will be regarded as a real logo.



(a) Training Stage

(b) Detection Stage

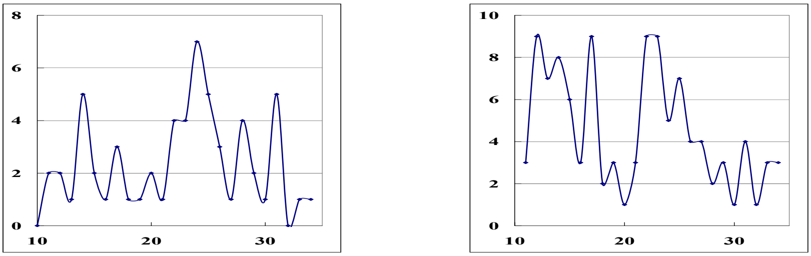
Fig. 2. Logo Detection Flowchart

3 Proposed Method

3.1 Shot Boundary Detection

A shot boundary detection tool [9]developed by the Tsinghua University is used for shot segmentation. It has three components: fade out/in (FOI) detector, cut detector and gradual transition detector. The standard deviation feature is utilized in FOI detection process. In the cut detector, the second order derivative method is used to boost the precision of cut candidates.The finite state automata model is adapted for the gradual transition detector.

In a half soccer video ($\sim$ 45 minutes), generally there are tens of logo sequences, some of the logo sequences can be segmented out as a separate shot, and labeled as GT by the shot boundary detection tool. A histogram of these GTs according to their durations is built. See Figure 3 for two examples. Multiple local peaks will occur. The highest peak is assumed to contain the logo



(a)

(b)

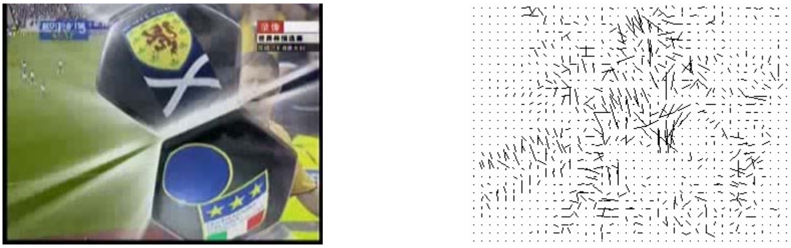
Fig. 3. Histogram of GT Duration in a Sport Video

sequences, and is fed into the alignment procedure. Select the remain highest peak, feed into the alignment procedure if a confident logo template has not been found before. Repeat above procedure until a high confident logo template is extracted successfully.

3.2 Calculate Motion Features

Color histogram has been used as the feature in the literature [7,8]. The result is not satisfy since the background is very complicate in a logo transition and the background changes from logo to logo. According to observation 4 above, the motion of a logo object is significant and generic. Here the optical flow feature is used.

In [10], Horn and Schunck use intensity-based difference features to calculate the feature [11]. In one frame, suppose an image point(x, y) at time t is moved to $(x + d_x, y + d_y)$ at time $t + d_t$, where the motion vector is denoted as (d_x, d_y). The Figure 4 is a frame in a logo template and the correspond optical flow field. In Figure 4 (b) The length of the black lines stand for the moving distance of a pixel, and the direction of the line is the moving direction of the pixel. From Figure 1(a) below and Figure 4(a), we can see that the logo is flying from left to right. Figure 4(b) demonstrates the corresponding optical flow field for frame in Figure 4(a).



(a) Original Frame

(b) Optical Flow Field

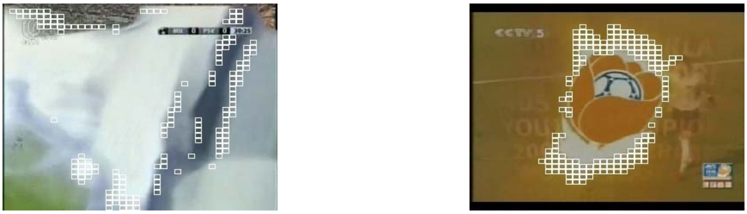
Fig. 4. Motion Vector Feature

Each frame in the system is reshaped to 320 × 240 for convenience. Each frame is then partitioned into 40 × 40 blocks uniformly. In each block, the optical flow features are computed for each pixel, and the magnitudes and dominant direction of each block is computed by accumulating on pixels in the block. See Equation (1), where I is the motion magnitudes and α_i is the direction of optical flow of a pixel i in the block. The direction histogram is formed with 16 bins. The direction that has the maximum number of pixels is chosen as the dominant direction. Thus each block has two parameters: one magnitude and one dominant direction.

$$I = \sum_{i \in block} \sqrt{d_{xi}^2 + d_{yi}^2} \qquad \alpha_i = \arctan(d_{yi}/d_{xi}) \tag{1}$$

3.3 Align the GT Sequences by Dynamic Programming

The similarity score $SC_{frame}(i, j)$ between frame i and j is defined as: select top 10% blocks that have the highest magnitudes in frame i and j respectively. If a block index is occurred in both top 10% lists, and both has the same dominant direction, the score add by 1. Figure 5 shows two example logo frames with the selected blocks. So the similarity of two frames is the proportion of common blocks that have the same position and same motion direction in two frames.The higher the score, the more similar of the two frames.



(a) AcMilan-Eindhoven Video

(b) Uklan-China Video

Fig. 5. The Top 10% Blocks of Motion Magnitudes

Sequence alignment is based on the dynamic programming. The Needleman-Wunsch-Sellers algorithm [12] [13] is a classic global dynamic programming and widely used. If the length of two sequences are M, N respectively, the substitution matrix is defined in Equation (2)

$$D_{i,j} = \max \begin{cases} D_{i-1,j-1} + SC_{frame}(i, j) \\ D_{i,j-1} + \omega \\ D_{i-1,j} + \omega \end{cases} \tag{2}$$

where $D_{0,0} = 0$, ω is the gap penalty, $SC_{frame}(i, j)$ is the match score between frame i and frame j. Usually $SC_{frame}(i, j)$ is $0.1 \sim 0.3$. Here ω is set to 0.

The final score SC_{seq} of two GT sequences $\mathbb{S}_i$ and $\mathbb{S}_j$ is defined by $SC_{seq}(\mathbb{S}_i, \mathbb{S}_j) = D_{M,N}/n$, where $D_{M,N}$ is the right-bottom value of the substitute matrix and n is the number of matches of two sequences.

3.4 Extract the Logo Template and Key Frame

After the sequence alignment, C_N^2 match scores between each pair of sequences is acquired. N is the number of sequences in consideration. Higher scores come from the matching of two logo sequences. Other alignments such as logo with non-logo, non-logo with non-logo, will produce lower score. These scores can be classified into two classes: high scores class and low scores class.

The Gaussian Mixture Model [14] with two components is used to describe the distribution of the scores. See Equation (3), where x is the score, the p_i is gaussian component weight. Each component λ_i is represented by a Gaussian distribution $\lambda_i = N(p_i, \mu_i, \sigma_i^2)$. EM algorithm is used to estimate the parameter.

$$g(x|\mathit{\Lambda}) = \sum_{i=1}^{n} p_i g_i(x) \tag{3}$$

According to the Maximum Likelihood, the decision rule for feature x can be expressed by Equation (4)

$$\begin{cases} L(x) > T & \text{if } x \in \lambda_1 \\ L(x) \leq T & \text{if } x \in \lambda_2 \end{cases} \tag{4}$$

where $L(x) = (x - \mu_2)^2/\sigma_2^2 - (x - \mu_1)^2/\sigma_1^2$, $T = \ln(\sigma_1^2/\sigma_2^2) + 2\ln(p_2/p_1)$.

In the logo class, the sequence with highest sum score with all other logo sequences is considered as the logo template $\mathbb{S}_1$. See Equation (5), where N is the number of logo sequences in consideration. Also the parameters of the Gaussian mixture model are acquired.

$$\mathbb{S}_1 \triangleq \arg\max_i \sum_{\substack{\mathbb{S}_i, \mathbb{S}_j \in logo \\ 1 \leq j \leq N, j \neq i}} SC_{seq}(\mathbb{S}_i, \mathbb{S}_j) \tag{5}$$

In the same time, the key frame which is a frame in the learned logo template is extracted according to the previous alignment results, the key frame has the highest alignment sum score with other logos. Also the difference threshold T_d and logo duration is learned. The difference threshold T_d is equal to the average difference value between the key frame and the correspond frames in other logo sequences. The duration of the logo template L is the average duration of all the sequences in logo class.

3.5 Detection

In the logo detection stage, the key frame is used as a filter to scan the whole videos and provide the candidate logos position, shown in Figure 2(b). This procedure can efficiently speedup the procedure and promote the recall.

The difference threshold T_d is acquired to evaluate the difference between the current frame and key frame. If $\sum_{H \in \mathbb{B}}(H_{key} - H_{cur}) < T_d$, the frame is considered as a candidate logo position, where H_{key} and H_{cur} are the hue component of key frame and current frame respectively. All the candidate logos are extended by logo duration L and the position that the key frame is in the logo template. Then the extended candidate logo sequence is aligned with logo template, if the similarity SC_{seq} score meet with the Equation (4), the candidate sequence is a real logo.

4 Experiments

Extensive experiments are conducted on 5 soccer games and 1 table tennis game and 1 NBA game, totally there are 6 different logos in these videos, see Figure 6. For each soccer videos, the first half is used as the training set and the second half is used for detection. For the table tennis and NBA game, the first and second rounds are used for training, and the total videos are for logo detection.

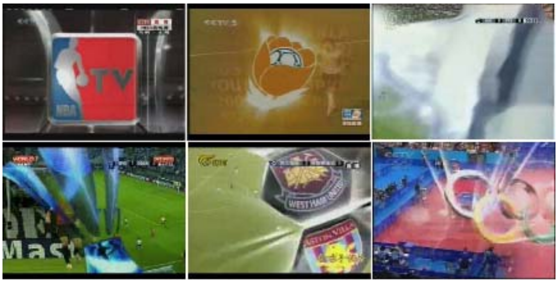


Fig. 6. Different Logos in the Experiments

For each video, two experiments were conducted: one for logo detection only within GTs that come from the shot detection tool, use only the logo template (it is computation infeasible to scan the whole video by the logo template); the other is performed on the whole video data by combining the key frame and logo template detection.

4.1 Results on GT Sequences Only

The results of the first experiment is shown in Table 1. We see the precision is very high, up to 100%, which demonstrates the effectiveness of our logo template alignment method. The recall is low, even 46.7% because many logo sequences has not been correctly segmented as GTs by shot detection tool. For the below 6 videos, the average recall is 77.4%.

4.2 Results on Whole Videos

The logo detection use both key frame and logo template is shown in Table 2. Again, the precision is perfect nearly 100% except in table tennis game. The recall is much higher than in Table 1, shown the detection that combining of key

Table 1. Results Detection Logos only on GT Sequences

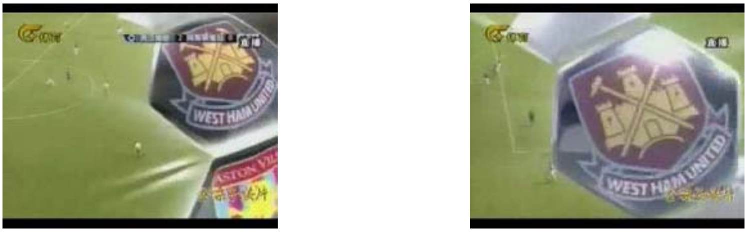
Test Video	Detect	Miss	False	Precision	Recall
AcMilan-Eindhoven	55	7	0	100%	88.7%
Arsenal-Ajax	28	2	0	100%	93.3%
Uklan-China	34	2	0	100%	94.4%
Manchestercity-Birmingham	14	16	0	100%	46.7%
Spur-Sun(NBA)	23	20	0	100%	53.5%
Waldner-Kong(Table Tennis)	45	11	0	100%	80.4%
Total	199	58	0	100%	77.4%

Table 2. Results Detection Logos only on Whole Videos

Test Video	Detect	Miss	False	Precision	Recall
AcMilan-Eindhoven	62	0	0	100%	100%
Arsenal-Ajax	28	2	0	100%	93.3%
Uklan-China	30	6	0	100%	83.3%
Manchestercity-Birmingham	26	4	0	100%	86.7%
Spur-Sun(NBA)	30	13	0	100%	69.8%
Waldner-Kong(Table Tennis)	56	0	1	98.2%	100%
Total	232	25	1	99.6%	90.3%

Table 3. Results on Two Different Logos in the Same Video

Result	Detect	Miss	False	Precision	Recall
on GT Sequences	9	39	0	100%	18.8%
on Whole Videos	31	17	0	100%	64.6%



(a) (b)

Fig. 7. Two Different Logos in the same video

frame and logo template is very efficient. For the below 6 videos, the average recall is 90.3%.

Results on the WestHamUnited-AstonVilla game is pretty bad, see Table 3. The recall is low in both cases because there are two different logos in the same video, shown in Figure 7.

5 Conclusions

In this paper, a novel sports logo detection method Based on motion analysis is proposed. The experiment results are satisfied. It has three advantages compare to previous methods: a logo template that contains a sequence of logo frames is used compare to only a single frame. The template can model the whole transition of the logo object; Dynamic programming is used to align two sequences. By clustering the scores of the dynamic programming, a logo template can be acquired automatically; The optical features are used to depict the motion characteristics of the logo object accurately. The whole system contains training and detection stages. In the training stage,a logo template and key frame is extracted. In the detection stage,the key frame is used to find the candidate logo positions and the logo template is used to verify. In both stages, dynamic programming is used. Experiments on different types of sports videos show that this method is effective and robust for detecting logos in sports videos.

However, the logo detector can not work well in some situations, for example, when multiple different logo objects occur in a same video, the performance will degrade greatly. This suggests that automatically detection of logos is still a problem far from being solved.

References

1. Nepal, S., Srinivasan, U., Reynolds, G.: Automatic detection of 'goal' segments in basketball videos. In: Proceedings of the ninth ACM international conference on Multimedia. (2001) 261–269
2. Tjondronegoro, D., Chen, Y.P.P., Pham, B.: The power of play-break for automatic detection and browsing of self-consumable sport video highlights. In: ACM SIGMM. (2004) 267–274
3. Kobla, V., DeMenthon, D., Doermann, D.: Detection of slow-motion replays for identifying sports videos. In: Proceedings of IEEE Workshop on Multimedia Signal Processing. (1999) 135 – 140
4. Pan, H., Van Beek, P., Sezan, M.: Detection of slow-motion replay segments in sports video for highlights generation. In: ICASSP. (2001) 1649–1652
5. Wang, L., Liu, X., Lin, S., Xu, G., Shum, H.Y.: Generic slow-motion replay detection in sports video. In: ICIP. (2004) 1585 – 1588
6. Pan, H., Li, B., Sezan, M.I.: Automatic detection of replay segments in broadcast sports programs by detection of logos in scene transitions. In: ICASSP. (2002) 3385–3388
7. Duan, L.Y., Xu, M., Tian, Q., Xu, C.S.: Mean shift based video segment representation and applications to replay detection. In: ICASSP. (2004) 709–712
8. Tong, X., Lu, H., Liu, Q., Jin, H.: Replay detection in broadcasting sports video. In: Proceedings - Third International Conference on Image and Graphics. (2004) 337–340
9. Yuan, J., Zhang, W., et al.: Tsinghua university at trecvid2004: Shot boundary detection and high-level feature extraction. In: TRECVID Workshops. (2004)
10. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artificial Intelligence **17**(1-3) (1981) 185–203

11. (http://www.intel.com/technology/computing/opencv/)
12. Needleman, S.B., Wunsch, C.D.: An efficient method applicable to the search for similarities in the amino acid sequences of two proteins. Journal of Molecular Biology **48** (1970) 444–453
13. Sellers, P.H.: An algorithm for the distance between two finite sequences. Journal of Combinatorial Theory **A16** (1974) 253–258
14. Dempster, A. P., L.N.M., D.B., R.: Maximum likelihood from incomplete data via the em algorithm. Journal of Royal Statistical Society **39B** (1977) 1–38

A Fast Selection Algorithm for Multiple Reference Frames in H.264/AVC*

Meng Qing-lei , Yao Chun-lian, and Li Bo

Digital Media Laboratory, School of Computer Science and Technology
Beihang University, Beijing 100083, China
mql198029@gmail.com

Abstract. The newest video coding standard H.264/AVC provides multiple reference frames motion estimation in the spatial region, and the optimal frame is selected by RDO (Rate Distortion Optimization) with high coding complexity. However, the coding efficiency only depends on the attribute of sequences, not on the number of reference frames. In this paper, statistical characteristics of the best reference frame with variable block size are studied, and a fast algorithm that takes into account the correlation is proposed. The reference frame of block mode may be chosen based on the computing result of the above block mode. Experimental results show that with similar Distortion performance, the algorithm can efficiently reduce the computational complexity by 19% averagely.

1 Introduction

The newest video coding standard H.264/AVC [1] is developed by the Joint Video Team (JVT) which was organized by ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) in 2001, which can typically outperforms all existing standards. H.264/AVC is similar to other standards such as MPEG-4 Video, which consists of a hybrid of temporal and spatial prediction, in conjunction with transform coding. But H.264 includes a number of new techniques such as variable block size, enhanced intra/inter prediction, 4×4 integer transform, adaptive in-loop deblocking filter, refined motion-compensated prediction, and new entropy coding, etc. Compared with the H.263 and MPEG-4(advanced simple profile), H.264/AVC can reduce 40%~50% bits-rate while keeping the equivalent video quality. However, the compression performance comes at a high computational cost [2].

In order to enhance the compression efficiency of P type frame, the motion estimation in H.264/AVC uses variable block size and multiple reference frames, which can greatly reduce prediction errors and obtain better performance. Reference software of H.264 adopts full search mode for each encode size block in every reference frame, and the optimal result is selected based on RDO, which contributes to heaviest computational load. To satisfy the requirement of real time, studying the fast algorithms how to reduce the code complexity becomes a key issue for specific encoder/decoder

* This work was supported by the NSFC (60573150), the National Defense Basic Research Foundation, and the Program for New Century Excellent Talents in University, and the research was made in the State Key Lab of Software Development Environment.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 458 – 465, 2006.

applications. Currently the research and implementation work mainly focus on mode decision process and achieves fairly good results. The main idea is as follows: forecasting the most coding mode based on the nature of sequences (Movement, venation, etc.); then using effective threshold value mechanisms for early withdrawal, which thereby reduces predictive modes and improves coding speed. In [3], candidate modes for current macroblock are first inferred from given coded adjacent macroblocks by adopting motion information and ratios of defined mode, and the final selection is made by a RDO approach. In [4], the threshold value is dynamically updated for each block mode in order to stop prediction quickly and correctly. Similar ideas are also explored in [5~6]. However, it can be seen that the computation is in proportion to the number of search frames, so it is necessary to reduce the multiple reference frames number. In [7~8], a fast motion estimation algorithm is proposed that takes into account the correlation of motion vectors in multiple frames, and a minor search windows is needed. [9] proposed a new idea that several conditions are used to decide whether it is necessary to search more reference frames. But algorithm simply adopt full search when the rest reference frame is beneficial.

A fast selection algorithm for multiple reference frames (FSAMR) in H.264/AVC is proposed in this paper, which reduces the encoding time by 19% averagely and can be combined with other methods such as [3~6] to further improve the speed. The paper is organized as follows: Section 2 briefly introduces the mode decision algorithm of multiple reference frames in H.264/AVC and gives the benefits of multiple reference frame prediction. In section 3, we analyze the statistical characteristics of the best reference frame among variable block size, and describe the details of our proposed fast algorithm for multiple reference frames selection. Finally, experiment results and concluding remarks are given in Section 4 and 5, respectively.

2 Overview of Multiple Reference Frames Prediction

2.1 Description of Multiple Reference Frames Prediction

H.264/AVC standard has extended the block based motion compensation by introducing tree structured variable-block size to approximate the shape of the moving objects within the MB more accurately. The size of a block can be 16×16 , 16×8 , 8×16 and 8×8 for motion compensation. In case 8×8 size is chosen, it can be further divided into smaller block size 8×4 , 4×8 and 4×4 . Besides the seven different sizes, an inter macroblock can also be coded in the Intra mode (Intra 4×4 and Intra16×16) and so-called SKIP mode. For this mode, neither a quantized prediction error signal, nor a motion vector or reference index parameter, has to be transmitted. H.264/AVC also supports multi-frame motion-compensated prediction. That is, more than one prior-coded frame can be used as a reference for motion-compensated prediction. The reference software of H.264/AVC JM94 performs full search to find the motion vector for each block in different sizes from previous one to five reference frame, shown as Fig.1 .

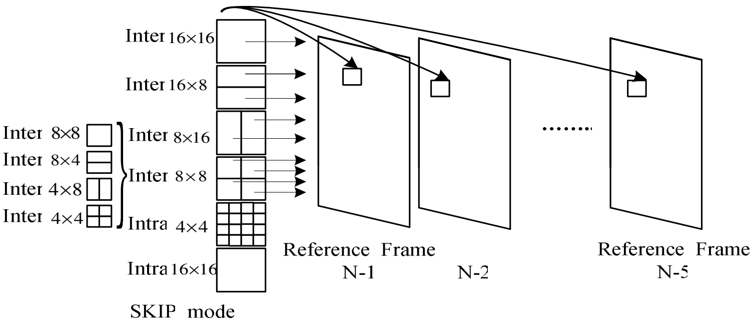


Fig. 1. Variable block size and multiple reference frames motion estimation

H.264/AVC selects the best mode and reference by using the RDO, which means that the final decision is made by minimizing the Lagrange formula (1):

$$J_{MODE} = D_{MODE}(ref) + \lambda \times R_{MODE}(ref) \tag{1}$$

Where ref being the reference frame position; J_{MODE} being R-D cost of the corresponding mode; λ being the Lagrange multiplier; R_{MODE} being the total bits-rate including the motion vectors, block mode, all transform coefficients, etc; D_{MODE} being the distortion between original frame and reconstructed frame.

The above scheme can gain better coding efficiency, but the RD cost of every mode and reference frame should be computed based on the actual rate and distortion, which are obtained only after compression and decompression. So the transform/inverse transform, quantization/inverse quantization, entropy coding have to be used repeatedly, and the complexity of computation is enhanced notably.

2.2 Benefits of Multiple Reference Frame Prediction

The video compression standard such as H.263 and MPEG-4, use single decoded frame as the reference frame, which can achieve better predictive results in most cases, except that there are some non-compensation regions in some special circumstances. However, multiple reference frames can gain better prediction result. The major reasons are described as follows, and the further details can be referred to [7,9]:

1) Due to the repetition of motion, objects or veins may have a better appearance at previous several frames than the latest reference frame.

2) Some parts of the objects or background may be covered by a moving object, which happens in many sequences, the hidden parts can not find a proper match in the latest reference frame, and may be found in the previous pictures when they were uncovered.

3) The shake and telescopic of the camera will lead to a rapid scene switch, and the same object position is in the difference reference frame.

4) Other reasons, why motion estimation of multiple reference frame get better performance than single reference, include the change of lighting and shadow, the sampling of picture, etc.

3 A Fast Multiple Reference Frames Selective Algorithm

3.1 Analyze of Multiple Reference Frames Prediction

In order to statistic and analysis of multiple reference frames motion estimation, we did some experiments on different sequences based on the H.264/AVC reference software JM94. we selected six typical sequences which are QCIF format (176×144) provided by MPEG standard: type A sequence, Mthr_dotr, Container with simple veins or slow movement; type B sequence, Foreman, Coastguard with middle veins or movement; type C sequence, Mobile, Bus with complex veins or intense movement. Table 1 lists the main parameters, and the conditions for all tests will be consistent.

Table 1. Test conditions

UseHadamard	On
SearchRange	16
SymbolMode	UVLC
ReferenceFrames	5
LoopFilter	On
AllMode	On
RDOptimization	On

We encoded 15 frames for each sequence, the structure of GOP (Group of picture) is IPPP (I type frame and P type frame), and the fixed QP is set to 28. Table2 shows the experimental result of coding efficiency with different reference frames. Compared with the single reference frame, ΔPSNR represents the benefits in luminance PSNR and ΔBits (%) denotes the percent of bit-rate variety with further reference frames. From Table2, we can conclude that the coding efficiency only depends on the nature of sequences, not on the number of reference frames. Generally, most benefits depend on the previous two reference frames obviously. We can also find that with the increased number of reference frames, the coding efficiency gain little, but the computational complexity increased sharply.

The mode decision result after motion estimation and intra prediction is also a very important cue [9].In Table3, in the expression A|B, A represents the possibility of a

Table 2. Comparison of the encode with different reference frame

	2 Reference	3 Reference	4 Reference	5 Reference
		ΔPSNR(dB)|ΔBits(%)		
Mthr_dotr	+0.048| +0.7	+0.059 | +0.9	+0.059|+0.9	+0.099|+1.7
Container	-0.028 | -6.1	+0.015 | -12.6	+0.025|-21.5	+0.049|-19.8
Foreman	+0.081| -3.0	+0.134 | -2.4	+0.152|-1.8	+0.174|-1.7
Coastguard	+0.082| +0.6	+0.092 | +0.5	+0.104|+1.8	+0.099|+1.4
Mobile	+0.069| -6.9	+0.129 |-13.5	+0.180|-16.5	+0.189|-19.0
Bus	+0.122| -2.6	+0.140 | -4.3	+0.158|-4.5	+0.177|-4.1

mode that can be chosen after the latest reference frame estimation, B is the possibility of A mode that can keep unchanged after 5 frame searched. Compared with [9], we added the analysis of SKIP mode. From Table3, we can see that: 53% of macroblocks need the latest reference frame; furthermore, when macroblock is split into smaller block size of 8×4, 4×8 and 4×4, there will be a better match on other reference frames; if macroblock adopt 16×16 mode or SKIP mode, there is simply circumstance and no further search is needed; Intra mode is seldom used. Summarily, the multiple reference gains better prediction for some special non-compensation region, and the efficiency depends on the nature of sequences such as veins and movement.

Table 3. Comparison of the encode with different reference frame

	SKIP	16×16	16×8	8×16	8×8	Intra
Mthr_dotr	51 | 90	19 | 54	11 | 73	08 | 49	09 | 62	2 |100
Container	80 | 95	10 | 53	03 | 31	05 | 31	02 | 44	0 | 0
Foreman	31 | 65	27 | 52	10 | 36	17 | 47	15 | 53	0 | 0
Coastguard	19 | 51	40 | 68	14 | 37	12 | 40	15 | 54	0 | 0
Mobile	06 | 25	27 | 42	11 | 28	10 | 34	46 | 58	0 | 0
Bus	08 | 51	36 | 54	17 | 41	10 | 39	28 | 78	1 | 77
Average	33 | 53	27 | 54	11 | 54	11 | 40	18 | 58	0 | 30
	33×53+27×54+11×54+11×40+18×58=53%					

Now we try to find out the correlation of variable block sizes. After motion estimation with 5 reference frames, we can get one best reference for 16×16 block mode, two best references for 16 × 8 block mode, and two best references for 8×16 block mode. From Table4 result, we can get some very useful information that about 84.5% blocks of 16 × 8 and 8×16 mode have the same reference frame which is consistent with 16×16 mode, and the percentage in 8 × 8 mode and further smaller block sizes is 89.8%.

Table 4. Correlation of the best reference frame among variable mode

	16×16 | 16×8 and 8×16	8×8 | 8×4 and 4×8
Mthr_dotr	88.8%	91.7%
Container	95.6%	97.5%
Foreman	81.2%	89.1%
Coastguard	86.7%	90.8%
Mobile	76.2%	83.5%
Bus	78.4%	85.9%
Average	84.5%	89.8%

3.2 Description of Fast Reference Frame Decision

Based on the above statistic and analysis, we propose a fast multiple reference frames selection algorithm for H.264/AVC, which is composed of the following steps:

Step 1: Perform the 16×16 block mode motion estimation referring to previous one to five reference frames, and obtain the best reference frame, noted as F_{16}.

Step 2: Do motion estimation on size of 16×8 and 8×16, and only the latest reference frame and F_{16} frame need to calculate the R-D cost.

Step 3: If 8×8 size is chosen, it can be further divided into smaller block size 8×4, 4×8 and 4×4, and a macroblock will loop four times for sub-macroblock mode. Perform the 8×8 block mode motion estimation for each reference frame, and obtain the best reference frame F_8 and R-D cost. $J_{8\times8}$, respectively.

Step 4: Calculate the R-D cost $J_{8\times4}$and $J_{4\times8}$ of the latest reference frame and F_8 frame.If $J_{8\times8} > J_{8\times4} / J_{4\times8}$, the 4×4 mode block search not only the latest reference frame, but also reference frames between F_8 and available furthest reference frame. Otherwise select the best reference between the latest reference frame and F_8.

Step 5: If all block mode have been processed, then process the next macroblock, otherwise jumps to step 3.

In the steps above, only the reference frame selection is modified during motion esti-mation, and it can be integrated with other fast algorithms to reduce complexity further.

4 Experimental Result and Discussions

The proposed FSAMR has been implemented based on JM94. The sequences and encoder condition are the same as shown in section 3.1. We encode 60 frames for each sequence, the structure of GOP is IPPP, the frame rate is 15f/s and the fixed QP value is set to 28. Peak signal noise ratio (PSNR), total motion estimation time, and total bits-rate of P-type frame are used as measurement. The results achieved by FSAMR and full search algorithm are presented in Table 5 and Figure 2. ΔPSNR represents the difference in luminance PSNR, ΔBits and ΔTime is defined as the formula (2):

$$Ratio = \frac{T_{pro} - T_{full}}{T_{full}} \times 100\% \qquad (2)$$

Where T_{full} and T_{pro} denote the result of full search and FSAMR, respectively.

Table 5. Comparison results

Sequence	ΔPSNR(dB)	ΔBits (%)	ΔTime (%)
Mthr_dotr	-0.022	+0.26	-20.8
Container	-0.022	-0.21	-20.0
Foreman	-0.037	+0.20	-19.2
Coastguard	+0.001	+0.17	-19.4
Mobile	-0.012	+1.70	-15.0
Bus	-0.024	+0.73	-19.6

As shown in Table5, our proposed algorithm reduces the computational complexity by 19%, meanwhile PSNR only decreases 0.02 slightly, and bits-rate increases only 0.47%, averagely. Besides, it can be seen that algorithm has a high content correlation between image sequences and FSAMR. Because the FSMAR algorithm uses statistical characteristics of the best reference frame among variable block size. Generally, simple veins or slow movement sequence has the stronger relativity, and this algorithm gains the better benefits; conversely, the effect of the fast reference frames selection algorithm may decrease.

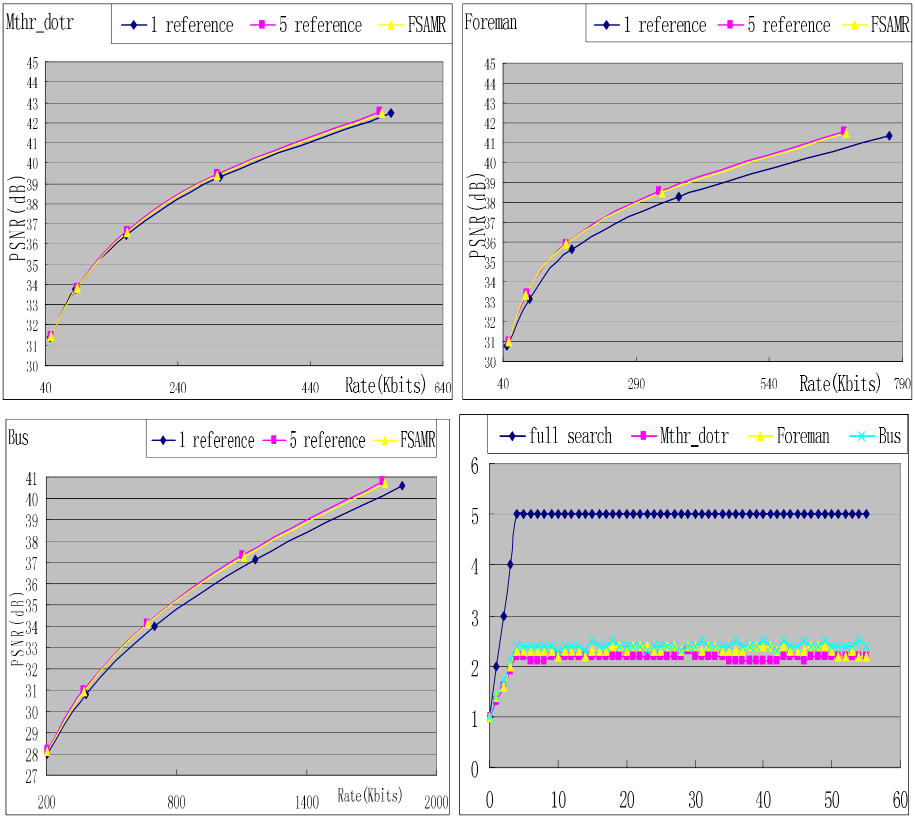


Fig. 2. Rate distortion curves and average searched frame

Fig.2 is the rate-distortion curves and average searched frames between 5 reference frames with full search, 1 reference frames with full search and 5 reference frames with FSAMR for different sequences. It is shown that compared with 5 reference frames with full search, FSAMR can efficiently reduce the number of searched reference frames with similar R-D, and the number of searched frames is no more than 3.

5 Conclusions

In this paper, we propose a new a fast selection algorithm, called FSAMR, for multiple reference frames in H.264/AVC. It is based on an analysis of statistical characteristics of the best reference frame among variable block size. The reference frame of block mode may be chosen based on the computing result of the above block mode. Experimental results show that compared with 5 reference frames search method, the algorithm can efficiently reduce the computational complexity, and meanwhile the degradation of the reconstructive video quality and the increase of the bits-rate are controlled under a reasonable level. Besides, this algorithm can be combined with other methods such as [3~6] to further improve the speed. How to perform a fast mode selection will also be our further work.

References

1. ISO/IEC FDIS 14496-10. Information technology-Coding of audio-visual objects Part 10: Advanced video coding[S]. Final Draft International Standard, 2003.
2. RavasiM, MattavelliM, Clerc C A. Computational Complexity Comparison of MPEG4 and JVT Codecs[S]. JVT-D153r1-L, Joint Video Team of ISO/IEC MPEG&ITU-T VCEG, Klagenfurt, Austria, 2002.
3. Zhu Hong, Wu Chengke, Fang Yong, A Novel Scheme for Fast Mode Decision within H. 264[J], ACTA Ectronica Sinica, 2005, 33(9):99-103(in Chinese).
4. Shen Gao, Tiejun Lun, An Improved Fast Mode Decision Algorithm in H.264 for Video Communications[A], ISSCAA2006 [C], Harbin,China,2006,57-60
5. KWON G, LEE J, YUN J, et al. Fast Inter-prediction method for mobile video communications using H.264/AVC[A]. International Conference on Consumer Electronics 2005[C]. Los Angeles, USA, 2005. 227-228.
6. Xiang Dong, Zhou Jingli, Yu Shengsheng, et al. Macroblock Coding Mode Predictive Method Based on Spatio-Temporal Correlation[J], Mini- Micro Systems ,2006, 27(1):101-103(in Chinese).
7. Ye-Ping Su, Ming-Ting Sun, Fast Multiple Reference Frame Motion Estimation for H.264/AVC[J], IEEE Transactions on CSVT, Accepted for future publication Volume PP, Issue 99, 2006 Page(s):1 - 1.
8. Mei-Juan Chen, Yi-Yen Chiang, Huang-Ju Li, Ming-Chieh Chi, Efficient Multi-Frame Motion Estimation Algorithms for MPEG-4 AVC/JVT/H.264[A], IEEE ISCAS 2004[C], Vancouver, Canada, May 2004,737-740
9. Yu-Wen Huang, Bing-Yu Hsieh, Tu-Chih Wang, Shao-Yi Chien, Analysis and Reduction of Reference Frames for Motion Estimation in MPEG-4 AVC/JVT/H.264[A], IEEE ISASSP 2003[C], Hong Kong, April 2003, 145-148

An Automotive Detector Using Biologically Motivated Selective Attention Model for a Blind Spot Monitor

Jaekyoung Moon[1], Jiyoung Yeo[2], Sungmoon Jeong[3],
PalJoo Yoon[4], and Minho Lee[1,2,3]

[1] Sensor Technology Research Center
[2] Dept. of Sensor Engineering
[3] School of Electrical Engineering and Computer Science, Kyungpook National University,
1370 Sankyuk-Dong, Puk-Gu, Taegu 702-701, South Korea
[4] Mando Corporation Central R&D Center,
413-5, Gomae-Ri, Giheung-Eub, Yongin-Si, Kyonggi-Do, South Korea
mholee@knu.ac.kr

Abstract. The conventional side-view and rear-view mirrors are not enough for driver's safety in an automobile. A driver may not be able to recognize the vehicle in a blind spot. In this paper, we propose an automotive detector algorithm using biologically motivated selective attention model for a blind spot monitor. This method decides a region of interest (ROI) which includes the blind spot from the successive image frames obtained by side-view cameras. It can detect the dangerous situations in the ROI using novelty points from the biologically motivated selective attention model, and alerts the driver whether there is dangerous object for changing the lane in driving. The proposed algorithm is based on deciding the ROI using difference from intensity histogram of a Gaussian smoothed image and finding the novelty points from the biologically motivated selective attention model. From variations of those novelty points, we determine whether a vehicle is approaching or not.

Keywords: Blind Spot Monitor; An Automotive Detector; Biologically Motivated Selective Attention Model.

1 Introduction

Automotive safety system has been advanced through the 21st century. The conventional auto safety technology was limited to passive purpose which was to protect occupants during a collision as seat belts and air bags. Recently, however, the passive safety system is combined with an active safety system which helps to avoid collisions. There are many examples of active safety system like anti-lock brakes and blind spot monitoring. The major causes of the worst car accidents mainly cause from the failure of drivers to stay within a lane. Therefore, active safety systems will be required to alert the driver before a collision happen when the driver attempts to change a lane without noticing the vehicle.

Vision-based technology has been used in order to improve automotive safety [1-3]. This system will be intelligent using advanced vision technologies including smart sensing [1]. Although the vision-based automotive detector relies on the performance

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 466 – 473, 2006.

of automotive-specific cameras, the image processing techniques are highly required to detect a dangerous situation with reliable performance. Most reliable ones are driving with an expert assistant who can give an alert signal whenever a driver attempts to change a lane without noticing an approaching vehicle. In this paper, we try to develop intelligent artificial assistant with human-like visual attention mechanism for giving alert signal in dangerous situation.

We propose a new algorithm to identify the vehicle in a blind spot using biologically motivated selective attention model. The human eye can focus on an attentive location in an input scene and select interesting visual information to process [6-8]. Considering the human-like selective-attention function, we determine a saliency map(SM) and several novelty points using bottom-up or task-independent processing. The bottom-up SM model generates plausible salient areas and novelty points using primitive features such as intensity, edge, and symmetry information. However, all of the novelty points in the SM may not be useful because we need to pay attention to the blind spot areas in driving. Therefore, we should consider the ROI decision method from the successive image frames. After deciding the ROI, we select meaningful novelty points within the ROI. As variations of those novelty points between successive two frames, we can determine whether a vehicle is approaching or not. We use a Euclidean distance and the longest path among the novelty points as a measure to alert the driver.

This paper is organized as follows; Section 2 describes a relation between a blind spot monitor and an ROI. Section 3 explains the proposed algorithm which consists of the SM and novelty points by biological background using the bottom-up SM model. Additionally, it includes the ROI decision method and measure to decide a dangerous situation. Section 4 shows the simulation results. Section 5 presents conclusion and discusses further works.

2 The Relation Between a Blind Spot and an ROI

We assume a freeway which has only cars and trucks with above 60km/h speed. A field of view (FOV) of the camera mounted a side-view mirror is about from 40 to 45 degrees and the FOV of the camera mounted a rear-view mirror about 120 degrees. In order to verify the automotive detector algorithm, we consider only the images from the camera of the side-view mirror until now. In addition, the proposed algorithm can process between 8 frames per second in real time from a camera. Cameras play an important role in vision-based intelligent safety systems. Side-view and rear-view enhancement is a common issue of a vision-based application. This paper focuses on a blind spot monitoring and warning of side-view. Thus, we mount side-view cameras of both side-view mirrors having about from 40 to 45 degrees of a FOV (field of view). For a camera to perform well in automotive applications, it must meet strict requirements. However, General-purpose cameras such as digital cameras, camcorders, and cell phone cameras are not well suited for use in automotive intelligent safety systems. The automotive camera must well perform in all conditions of intensity and direction of illumination, wavelengths of light in the scene, and speed of motion of the object being detected [4]. We use wide-VGA CMOS image sensor and global shutter robust the speed of motion.

A blind spot area is defined as shown in Fig. 1 [4]. When a driver plans to change lanes, area 1 can be seen by the driver through the side-view and the rear-view mirrors. And area 3 is directly visible area as the driver turning his/her head left. Additionally, a vehicle C can be seen by the side-view camera and by the driver through the side-view and the rear-view mirrors. However, area 2 and a vehicle B are the blind spot to be covered by the camera. The proposed algorithm only focuses on the blind spot area, whereas we do not consider a potential hazard. Therefore, we need to detect the vehicle B and area 2 in the blind spot using the side-view camera [4].

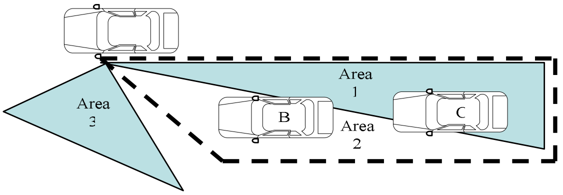


Fig. 1. Blind spot area from a side-view camera

Before an automotive detector algorithm uses the saliency map based on biologically motivated selective attention model, we decide a region of interest (ROI) from the frames of image obtained by the side-view camera. There are two reasons in deciding the ROI. One is to separate object from background of input image. Another is to find the saliency points within the ROI. This idea causes from the fact that there is hardly a difference between consecutive frames. A scenery and road can be a background. A vehicle can be separated as object. Therefore, we can only consider the saliency points inside the ROI without considering all of the input images. We employ a Euclidean distance and the longest length among the salient points as a measure to alert the driver.

3 An Automotive Detector Method

3.1 Saliency Map and Novelty Points Using a Visual Selective Attention Model for a Blind Spot Monitor

We use a visual selective attention model which is a biologically motivated bottom-up saliency map model. Fig. 2 shows the architecture of the bottom-up SM model. In order to model the human-like visual attention mechanism, we use the three bases of edge, intensity, and symmetry information, for which the roles of the retina cells, the LGN and the primary visual cortex are reflected in the previously proposed attention model [6-8]. In order to consider the shape information of an object, we consider the symmetry information. The symmetry information is obtained by the noise tolerant general symmetry transform (NTGST) method. Three feature maps are obtained by the following equations:

$$I(c, s) = |I(c) \ominus I(s)| \tag{1}$$

$$E(c, s) = |E(c) \ominus E(s)| \tag{2}$$

$$S\,(c, s) = |S\,(c) \ominus S\,(s)| \tag{3}$$

where "$\ominus$" represents interpolation to the finer scale and point-by-point subtraction. Totally, 18 feature maps are computed because the three feature maps individually have 6 different scales [6, 7]. Feature maps are combined into three "conspicuity maps," as shown in Eq. (4) where $\overline{I}$, $\overline{E}$, and $\overline{S}$ stand for intensity, edge, and symmetry, respectively. These are obtained through across-scale addition "$\oplus$" [7].

The feature maps ($\overline{I}$, $\overline{E}$, and $\overline{S}$) are constructed by center surround difference and normalization (CSD & N) of the three bases, which mimics the on-center and off-surround mechanism in our brain.

$$\begin{aligned} \overline{I} &= \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} N(I(c,s))\,, \\ \overline{E} &= \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} N(E(c,s))\,, \\ \overline{S} &= \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} N(S(c,s)) \end{aligned} \tag{4}$$

The saliency map is simply computed by summation of the feature maps for every location.

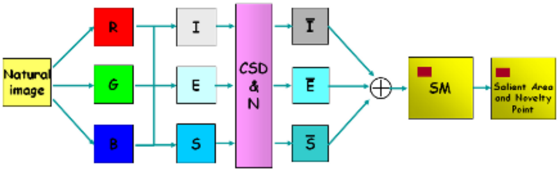


Fig. 2. The architecture considering both the bottom-up selective attention model, I: intensity image, E: edge image, S: symmetry image, CSD & N: center-surround difference and normalization, $\overline{I}$: intensity feature map, $\overline{E}$: edge feature map, $\overline{S}$: symmetry feature map, SM: saliency map, SP : saliency point, the small square block of feature maps : saliency area.

3.2 Decision of Region of Interest (ROI)

All of the novelty points in the SM may not be useful because we need to pay attention to the blind spot areas in driving. Therefore, we should consider the ROI decision method from the successive image frames. The proposed algorithm is based on deciding an ROI using difference from intensity histogram of Gaussian smoothed images.

Fig. 3 shows the ROI decision processor. In order to reduce noises of input images, we use a Gaussian filter. This processor may process about 8 frames per second. A vehicle speed is above 60km/h. We divide the Gaussian smoothed input images into

20 by 10 small block images and find an intensity histogram for each block to see the intensity variation of the local area. When the background scenery and road only change, the value of the intensity difference between frames can be very small. However, if the novelty, a vehicle approaches, the value of the intensity difference is larger than that when no objects appear. Thus, we need to consider the variation of intensity between successive frames. In order to find an appropriate value for each block, we include three processing, sliding-intensity histogram, quantization and mean operator for each block. Sliding-intensity histogram processor moves each block to column direction overlapping by fifty percentages and obtain the value of intensity histogram. Fifty percentages are obtained by trial and error using computer simulations. After finishing the sliding-intensity histogram processing to column direction, the processor moves it to row direction as same as to column direction. Because of each block with overlapping, the number of intensity histogram for blocks increases. Overlapping each block helps the ROI decision more accurate. The second processor, quantization processor decides an average value of the numbers of pixels for each block according to a quantization level. Eq. (5) represents quantization processing.

$$N_{\tau,b,l=0}(Q_l \le I(x,y) < Q_{l+1}) = \frac{\sum_{i=Q_l*256/Q_L}^{Q_l+256/Q_L} h_{\tau+\Delta t,b}(i) - h_{\tau,b}(i)}{256/Q_L} \tag{5}$$

where quantization level is divided into 16 level, Q_L is 16 and $h_{\tau,b}$ means intensity histogram in b block at τ frame. $N_{\tau,b,l=0}$ means an average value of intensity histogram at quantization level $l = 0,1,2,\cdots,15$, and τ frame for *I(x,y)*, a pixel intensity range from Q_l to Q_{l+1}. After quantization processor, we need to obtain the difference of the quantized value between two successive frames in order to know the intensity variation. Eq. (6) represents the difference of the quantized value between two successive frames at quantization level *l*, *b* block, τ and $\tau + \Delta t$ frames. Then, we find mean value for each block as shown in Eq. (7).

$$\begin{aligned} &Q_{\tau+\Delta t,b,l}(i) \\ &= N_{\tau+\Delta t,b,l}(Q_l \le I(x,y) < Q_{l+1}) - N_{\tau,b,l}(Q_l \le I(x,y) < Q_{l+1}) \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad \textit{for} \;\; l = 0,1,2,\cdots,15 \end{aligned} \tag{6}$$

$$M_b = \frac{\sum_{l=0}^{Q_L} Q_{\tau+\Delta t,b,l}(i) - Q_{\tau,b,l}(i)}{Q_L} \tag{7}$$

where, $Q_{\tau+\Delta t,b,l}$ represents the difference of the quantized value between two successive frames between $N_{\tau+\Delta t,b,l}$ and $N_{\tau,b,l}$ for $l = 0,1,2,\cdots,15$. We need the representative value from $Q_{\tau+\Delta t,b,l}$, $l = 0,1,2,\cdots,15$ for each block using an average operation. M_b means an average value. From all mean values for all blocks, we decide a threshold whether blocks in $\tau+\Delta t$ frame is the ROI or not. The threshold is obtained by choosing a median of all mean values. If the mean value of the block is smaller than the threshold, the block can not be the ROI. Otherwise, the block may be the ROI. However, the overlapping part is preferentially chosen by a value decided not for the ROI.

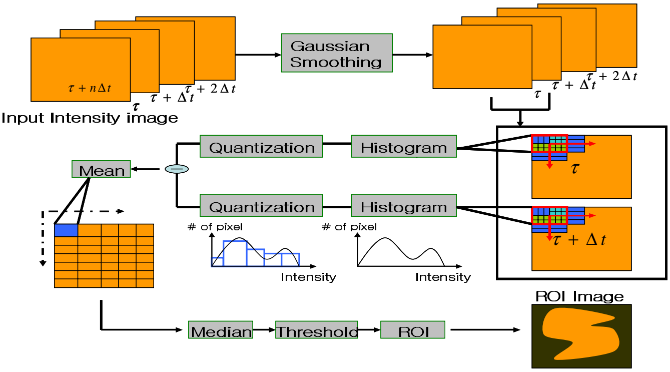


Fig. 3. The proposed ROI based histogram model

Fig. 4 shows simple sketch for novelty points within the ROI. This process applies the ROI to SM in order to determine meaningful novelty points. Fig. 4 (a) represents that the novelty points within the ROI are only determined by the variation of background. Fig. 4 (b) shows that the novelty points within the ROI are determined by the intensity variation due to a vehicle's appearance. If the vehicle is approaching, the ROI extends to wider area including a blind spot. Otherwise, the ROI is distributed in many places in image with smaller area. Finally, we need to determine when we give a warning signal of a dangerous situation to driver.

There can be various methods deciding whether there is a dangerous element in the ROI or not. We use Euclidian distances considering distribution information of novelty points. In case of approaching a dangerous object such as a car and a truck

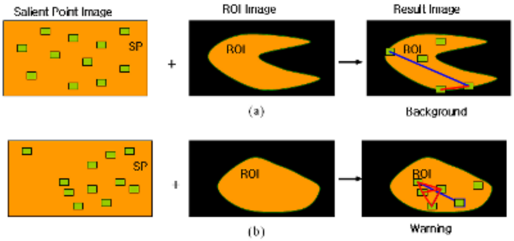


Fig. 4. (a) Novelty points and ED within the ROI when there is no a vehicle. (b) Novelty points and ED within the ROI when there is a vehicle.

in the ROI, novelty points are closely distributed in the object because the dangerous object may be more salient than background. On the other hand, in case of not existing a dangerous object, novelty points are evenly distributed in the ROI and are relatively far from each novelty point. From the fact, we use two measures, of which the first is to check the longest distance between novelty points in the ROI is less than a threshold, and the second is to check whether the number of novelty points within a predefined distance is above three, because the novelty points in ROI are denser in dangerous situation than that in normal situation. Using the proposed algorithm, we can give the warning signal to a driver through the blind spot monitor.

4 Computer Simulation and Experimental Results

In order to verify the automotive detection algorithm, we simulate and analyze one case of images. The case has a truck in a blind spot. Fig. 5 shows the result of an automotive detector using an ROI decision method combined with a visual selective attention model for a truck. Fig. 5 (a) shows input images for a truck. We process a Gaussian smoothing filter in input frames to reduce noises and decide the ROI using intensity histogram based on blocks as shown in Fig. 5 (b) and (c). Fig. 5 (d) and (e) show the saliency map and the novelty points using a visual selective attention model, respectively. Then we choose only the novelty points included in the ROI as shown in Fig. 5 (f). We consider only frames having more than three novelty points. For the truck images, we can consider τ , $\tau + 2\Delta t$ and $\tau + 7\Delta t$. When we find Euclidian distances (ED) of all cases among novelty points for chosen frames, we see two frames, at τ and $\tau + 2\Delta t$ have only long paths having about over ED 100. However, in frame, at $\tau + 7\Delta t$, there are more than three short paths having about ED 20~30. Therefore, we can give the warning signal at the frame at $\tau + 7\Delta t$ to driver.

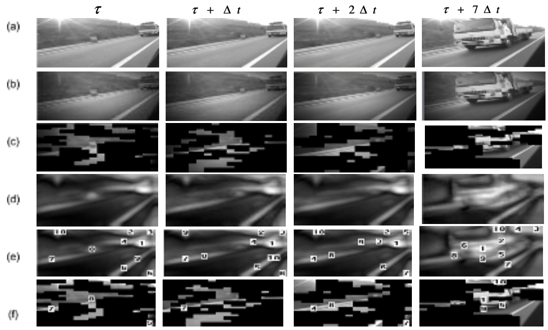


Fig. 5. The result of an automotive detector using an ROI decision method combined with a visual selective attention model: (a) input images, (b) Gaussian smoothed images (c) the ROI images using intensity histogram based on blocks (d) the saliency map using a visual selective attention model (e) novelty points in the saliency map (f) Novelty points within the ROI

5 Conclusions

We propose an automotive detection algorithm using a biologically motivated selective attention model for a blind spot monitor. This method decides an effective ROI which includes the blind spot from the successive image frames obtained by side-view cameras. It can detect the dangerous situations in the ROI using novelty points from the biologically motivated selective attention model, and alerts the driver whether there is dangerous for changing the lane in driving. The proposed algorithm is based on deciding the ROI using difference from intensity histogram of a Gaussian smoothed image and finding the novelty points from the biologically motivated selective attention model. From variations of those novelty points, we determine whether a vehicle is approaching or not. From simulation results, we can verify the proposed method detects the hazardous situation from input images.

Acknowledgement

This research is supported by Components and Materials Technology Development Program, VISS(Vision-based Intelligent Steering System) Project. Specifically, this research is commissioned by MANDO's sub-project IPAS(Intelligent Parking Assistant System).

References

1. Katz, D., Lukasiak, T., and Gentile, R.: Use of Video Technology To Improve Automotive Safety Becomes More Feasible with Blackfin™ Processors, Analog Devices, http://www.analog.com/analogdialogue
2. Furukawa, Y.: Overview R&D on Active Safety in Japan, Shibaura Institute of technology
3. Mota, S., Ros, E., Ortigosa, E. M., and Pelayo, F. J.: Bio-inspired Motion Detection for a Blind Spot Overtaking Monitor, International Journal of Robotics and Automation, vol. 19 (2004)
4. Automotive Cameras for Safety and Convenience Applications - White Paper by SMaL Camera Technologies, Inc. (2004) ver. 1
5. Rasshofer, R. H., and Gresser, K.: Automotive Radar and Lidar Systems for Next Generation Driver Assistance Functions, BMW Group Research and Technology, Germany
6. Park, S. J., Shin, J. K., and Lee, M.: Biologically inspired saliency map model for bottom-up visual attention, Lecture Notes in Computer Science, vol. 2525 (2002) 418-426
7. Itti, L., Koch, C., and Niebur, E.: A model of saliency-based visual attention for rapid scene analysis, IEEE Trans. Patt. Anal. Mach. Intell. vol. 20, no. 11 (1998) 1254-1259
8. Navalpakkam, V., and Itti, L.: A goal oriented attention guidance model, BMCV 2002, Lecture Notes in Computer Science, vol. 2525 (2002) 472-479

Wavelet Energy Signature: Comparison and Analysis

Xiaobin Li[1] and Zheng Tian[2]

[1] Department of Applied Mathematics, Northwestern Polytechnical University,
Xi'an, 710072, China
lixiaobin2006@gmail.com
[2] Department of Applied Mathematics, Northwestern Polytechnical University,
Xi'an, 710072, China

Abstract. Though wavelet transform based methods have recently raised increasing interests in texture analysis due to their good space and frequency localization, many issues related to the choice of the wavelet basis and texture feature remain unresolved. In this paper, we evaluate the performance of seven wavelet energy signatures and eight wavelet basis for texture discrimination. Experimental results on 111 Brodatz textures show that the feature extracted from high and middle frequency channels is more suitable for texture analysis and the choice of wavelet basis has some influence on texture discrimination.

1 Introduction

Texture analysis has played an important role in many areas including robotic vision, industrial monitoring, remote sensing, assisted medical diagnosis and automated target recognition. There are three primary issues in texture analysis, such as texture classification, texture segmentation and synthesis. Extracting textural features is the main step for analyzing texture.

Many features extraction techniques have been invented in the past for texture analysis, such as features based on gray level co-occurrence matrix [1], features based on run length matrix[2] and singular value decomposition spectrum[3], features based on Gaussian Markov random fields (GMRF) [4] and Gibbs random fields[5] and features based on local linear transformations [6] etc. These methods above are usually restricted to the analysis of spatial interactions over relatively small neighborhoods on a single scale. However, psychovisual studies indicate that the human visual system processed images in a multiscale way and an important aspect of texture is scale [7]. So, as a result, more recently methods based on multi-resolusion or multi-channel analysis such as Gabor filters [8], [9] and wavelet transform [10~13] have received a lot of attention. Though the Gabor filter is famous for its simulation with human vision, the output of Gabor filter banks are not mutually orthogonal, which may result in a significant correlation between textures. Moreover, these transformations are usually not reversible, which limits their applicability for texture synthesis. As a preferred tool for multiresolution analysis, wavelet theory provides a more formal, solid and unified approach to multiresolution representation [14], [15].

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 474 – 480, 2006.

Many wavelet transform based features have been invented. Among them are wavelet energy signature (WES) which is the most popular feature used in wavelet texture analysis [12]. Despite the empirical success, the choices of wavelet basis (WB) and WES remain unsolved. The impact of the WB has been partially addressed in recently published papers. For example, in [16], Chang and Kuo have suggested that the filter selection has little information on the texture classification. But, on the other hand, the experiments in [17], [18] imply that it is an important issue the choice of filter bank in the wavelet texture characterization. In this paper we analyze the performance of seven WESs, which are combinations of features extracted from different frequency bands, and eight WBs on 111 Brodatz textures [19]. The primary aim is to investigate which frequency bands play an important role in texture description and whether the choice of WB can influence the texture discrimination. This paper is organized as follows. Section 2 presents the basic concept of the wavelet transform. Section 3 gives the methodology and experiment results. Conclusions are given in section 4.

2 Wavelet Transforms

The wavelet transform performs the decomposition of a signal f with a family of function $\psi_{m,n}(x)$ obtained through translation and dilation of a kernel function called mother wavelet via

$$\psi_{m,n}(x) = 2^{-m/2}\psi(2^{-m}x - n)\,. \tag{1}$$

The mother wavelet can be constructed from two-scale difference equations

$$\varphi(x) = \sqrt{2}\sum_{k} h(k)\varphi(2x - k)\,, \tag{2}$$

$$\psi(x) = \sqrt{2}\sum_{k} g(k)\varphi(2x - k)\,, \tag{3}$$

where $\varphi(x)$ is called scaling function , and $h(k)$ and $g(k)$ can be viewed as filter coefficients of half band low-pass and high-pass filters, respectively.

The filter coefficients $h(k)$ and $g(k)$ play a very crucial role in discrete wavelet transform (DWT) and they can be used for DWT computation instead of the explicit forms for $\varphi(x)$ and $\psi(x)$. In fact, a J -level wavelet decomposition can be written as

$$f_0(x) = \sum_{k} c_{0,k}\varphi_{0,k}(x) \tag{4}$$

$$= \sum_{k}(c_{J+1,k}\varphi_{J+1,k}(x) + \sum_{j=0}^{J} d_{J+1,k}\psi_{J+1,k}(x))\,, \tag{5}$$

where coefficients $c_{0,k}$ are given and

$$c_{j+1,n} = \sum_{k} c_{j,k} h(k - 2n)\,, \tag{6}$$

$$d_{j+1,n} = \sum_{k} d_{j,k} g(k - 2n)\,. \tag{7}$$

The above two formulas provides a recursive algorithm for wavelet decomposition through filter coefficients $h(k)$ and $g(k)$. The final output of DWT of a signal include a set of detail coefficients $d_{j,k}$ and approximation coefficients $c_{j,k}$.

A two-dimensional DWT can be treated as two one-dimensional transforms over image rows and columns separately. This will generate three orientation selective detail subimages $D_j^{(k)}$, $k = h, v, d$ and an approximate subimage A_j where j denotes the decomposition level. The process then repeated on the approximate subimage to produce the next level of the resolution. Figure 1 shows a two-level hierarchical decomposition.

Since textures, either micro or macro, have non-uniform gray level variations, they are statistically characterized by the features derived from transformed coefficients in approximation and detail subimages. In other words, we can use these features to analyze the texture.

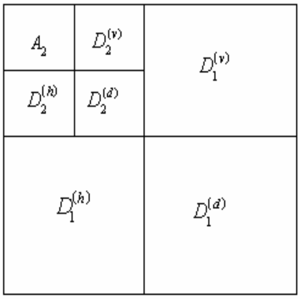


Fig. 1. Wavelet representation of image by detail subimages and approximate subimage

3 Comparison and Analysis

Wavelet texture analysis is considered to be the current state of the art among other texture analysis methods and has shown better performance than other methods in many cases. In this section, we evaluate the performance of seven WESs and eight WBs by using 111 Brodatz textures, each with a size of 75 × 75 pixels and 256 gray levels. Fig. 2 illustrates some textures from our experimental set. The eight WBs are Haar wavelet, Db2 wavelet, Db4 wavelet, Db7 wavelet, Coif2 wavelet, Bior 2.6 wavelet and Dmey wavelet.

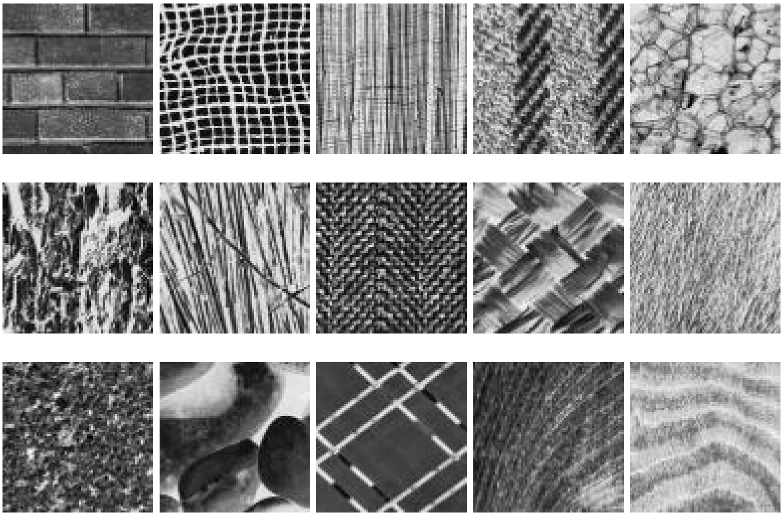


Fig. 2. Some textures from the experimental set

Table 1. Seven wavelet energy signatures

F$_1$	$\left(\frac{\lVert A_2\rVert_F^2}{ared(A_2)}, \frac{\lVert D_2^{(h)}\rVert_F^2}{ared(D_2^{(h)})}, \frac{\lVert D_2^{(v)}\rVert_F^2}{ared(D_2^{(v)})}, \frac{\lVert D_2^{(d)}\rVert_F^2}{ared(D_2^{(d)})}, \frac{\lVert D_1^{(h)}\rVert_F^2}{ared(D_1^{(h)})}, \frac{\lVert D_1^{(v)}\rVert_F^2}{ared(D_1^{(v)})}, \frac{\lVert D_1^{(d)}\rVert_F^2}{ared(D_1^{(d)})}\right)$
F$_2$	$\left(\frac{\lVert D_1^{(h)}\rVert_F^2}{ared(D_1^{(h)})}, \frac{\lVert D_1^{(v)}\rVert_F^2}{ared(D_1^{(v)})}, \frac{\lVert D_1^{(d)}\rVert_F^2}{ared(D_1^{(d)})}\right)$
F$_3$	$\left(\frac{\lVert D_2^{(h)}\rVert_F^2}{ared(D_2^{(h)})}, \frac{\lVert D_2^{(v)}\rVert_F^2}{ared(D_2^{(v)})}, \frac{\lVert D_2^{(d)}\rVert_F^2}{ared(D_2^{(d)})}, \frac{\lVert D_1^{(h)}\rVert_F^2}{ared(D_1^{(h)})}, \frac{\lVert D_1^{(v)}\rVert_F^2}{ared(D_1^{(v)})}, \frac{\lVert D_1^{(d)}\rVert_F^2}{ared(D_1^{(d)})}\right)$
F$_4$	$\left(\frac{\lVert D_1^{(h)}\rVert_F^2}{ared(D_1^{(h)})}, \frac{\lVert D_1^{(v)}\rVert_F^2}{ared(D_1^{(v)})}, \frac{\lVert D_1^{(d)}\rVert_F^2}{ared(D_1^{(d)})}, \frac{\lVert D_2^{(d)}\rVert_F^2}{ared(D_2^{(d)})}\right)$
F$_5$	$\left(\frac{\lVert A_2\rVert_F^2}{ared(A_2)}, \frac{\lVert D_1^{(h)}\rVert_F^2}{ared(D_1^{(h)})}, \frac{\lVert D_1^{(v)}\rVert_F^2}{ared(D_1^{(v)})}, \frac{\lVert D_1^{(d)}\rVert_F^2}{ared(D_1^{(d)})}\right)$
F$_6$	$\left(\frac{\lVert A_2\rVert_F^2}{ared(A_2)}, \frac{\lVert D_2^{(h)}\rVert_F^2}{ared(D_2^{(h)})}, \frac{\lVert D_2^{(v)}\rVert_F^2}{ared(D_2^{(v)})}, \frac{\lVert D_2^{(d)}\rVert_F^2}{ared(D_2^{(d)})}\right)$
F$_7$	$\left(\frac{\lVert D_2^{(h)}\rVert_F^2}{ared(D_2^{(h)})}, \frac{\lVert D_2^{(v)}\rVert_F^2}{ared(D_2^{(v)})}, \frac{\lVert D_2^{(d)}\rVert_F^2}{ared(D_2^{(d)})}\right)$

3.1 Texture Features Selection

The two-level DWT is firstly applied to the texture image. This generates six detail subimages and one approximation subimage. Then the normalized energy of each subimage is calculated and some of them are employed as elements of the texture feature vector. In our test, we choose seven WESs which are given in table I, where $\|\bullet\|_F$ denotes the Frobenius norm and $area(\bullet)$ denotes the product of row number and column number of a matrix.

3.2 Performance Evaluation

For every WB, firstly, we select randomly 20 texture images from 111 Brodatz texture images. Then we extract feature vector $F_i(i = 1, 2, \cdots, 6)$ from each texture image. For F_i, this results in 20 vectors. The cosine of angle of every two of 20 vectors is computed and 190 values are got. Finally, the mean and variance of these 190 values, denoted by $mean(F_i)$ and $\mathrm{var}(F_i)$, are calculated to show the performance of feature F_i. At the same time the best feature for every WB is given. In our experiments, since the variation of seven $\mathrm{var}(F_i)s$ is small, we choose the feature corresponding to the minimal $mean(F_i)$ as the best choice for every WB. To derive some significant statistics, this experiment was repeated 100 times. Table II shows the experimental results, Where $Angle = \arccos(\min_{1\le i\le 6}\{\frac{1}{100}\sum mean(F_i)\})$. In 100 experiments, a surprising thing is for every WB the best feature is same at each time, so Table II also shows the best feature for every wavelet.

From the experiment results, one thing is obvious that for eight WBs the best features are all F_4 which extracted from the detail subimages $D_1^{(j)}(j = h, v, d)$ and $D_2^{(d)}$. This shows that the texture characteristic are mainly in high and middle frequency regions. The other thing is $Angle$ s for eight WBs all lie in the interval $[34^o, 40^o]$, this shows the ability of WB for texture discrimination. If set

$$MaxA = \max\{Angle\}, MinA = \min\{Angle\}, \tag{8}$$

then

$$\frac{MaxA - MinA}{\frac{1}{8}\sum Angle} = 0.1072\,. \tag{9}$$

This shows that in wavelet texture characterization the choice of WB could affect the texture discrimination. Especially, in eight WBs, Haar wavelet is the most unsuitable for texture discrimination and in contrast Db7 wavelet is the best.

Table 2. The experimental results

WB	Haar	Db2	Db4	Db7
Angle (degr ee)	35.0888	36.8757	38.4868	39.1391
Feature	F4	F4	F4	F4
WB	Sym8	Coif2	Bior2.6	Dmey
Angle (degr ee)	38.3809	38.8173	36.6914	38.8966
Feature	F4	F4	F4	F4

4 Conclusions

In this paper we evaluate the performance of seven WESs and eight WBs for texture discrimination. Our experiment results show that in the wavelet texture characterization the choice of WB could influence the texture discrimination. Our findings, that feature F_4 is more suitable for texture analysis than other six features which are used in many other studies, show that the texture characteristic are mainly in high and middle frequency regions. This result can be used for feature selection in the design of system for texture description and synthesis and other areas, such as image coding.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (60375003) , Aeronautics and Astronautics Basal Science Foundation of China (03I53059).

References

1. Haralick, R.M., Shanmugam, K.K., Dinstein, L.: Features for Image Classification. IEEE Trans. Syst. Cyb. 8 (6) (1973) 610–621.
2. Galloway, M. M.: Texture Analysis Using Gray Level Run Lengths," Comput. Graphics Image Process. Vol. 4 (1975) 172–179.
3. Ashjari, B.: Singular Value Decomposition Texture Measurement for Image Classification, PhD, thesis, University of Southern Califomia, Los Angeles, CA, (1982).
4. Cross, G.R., Jain, A.K.: Markov Random Field Texture Models. IEEE Trans. Pattern Anal. Machine Intell. PAMI-5(1) (1983) 25–39.
5. Derin, H., Elliot, H.: Model and Segmentation of Noisy and Textured Images Using Gibbs random Fields. IEEE Trans. Pattern Anal. Machine Intell. PAMI-1 (1987) 251-259.
6. Unser, M.: Local Linear Transforms for Texture Measurements. Signal Process. Vol. 11 (1986) 61–79.

7. Daugman, J.G.: An Information-theoretic View of Analog Representation in Striate Cortex. Comp. Neurosc. (1990) 403-424.
8. Bovik, A. C., Clark, M., and Geisler, W. S.: Multichannel Texture Analysis Using Localized Spatial Filters. IEEE Trans. PAMI, Vol. 12 (1990) 55-73.
9. Jain, A. K. and Farrokhnia, F.: Unsupervised Texture Segmentation Using Gabor Fillters. Pattern Recognition, Vol. 24 (1991) 1167-1186.
10. Arivazhagan, S. and Ganesan, L.: Texture Classification Using Wavelet Transform. Pattern Recogn. Lett. Vol.24 (2003) 1513–1521.
11. Arivazhagan, S. and Ganesan, L.: Texture Segmentation Using Wavelet Transform. Pattern Recogn. Lett. Vol. 24 (2003) 3197–3203.
12. Bharati, M. H., Liu, J. J. and Macgregor, J. F.: Image Texture Analysis: Methods and Comparisons. Chemometrics and intelligent laboratory systems, Vol. 72 (2004) 57-71.
13. Mor, E. and Aladjem, M.: Boundary Refinements for Wavelet-domain Multiscale Texture Segmentation. Image and Vision Computing, Vol. 23 (2005) 1150-1158.
14. Mallat, S.: A Theory for Multiresolution Signal Decomposition: the Wavelet Representation. IEEE Trans. PAMI, Vol.11 (7) (1989) 674-693.
15. Daubechies, I.: Ten Lectures on Wavelets. Philadelphia, PA: SIAM (1992).
16. Chang, T. and Kuo, C.-C. J.: Texture Analysis and Classification with Tree-structured Wavelet Transform. IEEE Trans. Image Processing, Vol. 2(4) (1993) 429-441.
17. Unser, M.: Texture Classification and Segmentation Using Wavelet Frames. IEEE Transactions on Image Processing, Vol.4 (11) (1995) 1549-1560.
18. Mojsilović, A., Popović, M. V. and Rackov, D. M.: On the Selection of an Optimal Wavelet Basis for Texture Characterization. IEEE Transactions on Image Processing, Vol.9 (12) (2000) 2043-2050.
19. Brodatz, P.: Textures, a Photographic Album for Artists and Designers. Dover Publications, New York (1966).

Image Fusion Based on PCA and Undecimated Discrete Wavelet Transform

Wei Liu, Jie Huang, and Yongjun Zhao

Information Science and Technology Institute, Zhengzhou, Henan, 450002, China
buffler@163.com

Abstract. On the basis of analyzing the performances of popular image fusion methods, a new remote sensing image fusion method based on principal component analysis (PCA), high pass filter (HPF) and undecimated discrete wavelet transform (UDWT) is proposed. Some measure parameters are suggested to evaluate the fusion method. Experiments have been performed with the SPOT panchromatic image and the TM multi-spectral image. Both subjectively qualitative analysis and objectively quantitative evaluation verify the performance of the new method. With the same wavelet transform level, the fusion image using the proposed method preserves more sophisticated spatial details and distorts less spectral information in comparison with the fusion image using the traditional discrete wavelet transform (DWT) method.

1 Introduction

By the organic integration of various and complementary information, multi-sensor data fusion can furthest utilize multi-resource information and reduce the uncertainty or error of interpretation with the single resource, thereby greatly enhance the effectiveness of features extraction, classification, target detection, identification, etc.

Multi-spectral and panchromatic images are two kinds of data commonly used. Multi-spectral images contain abundant spectral information, but have poorly performance of the spatial details because of lower resolution. Panchromatic images have rich spatial details. The purpose of fusion is to maintain spectral information of multi-spectral images and improve the spatial details at the same time.

The classical multi-spectral and panchromatic imagery fusion methods include the High Pass Filter (HPF) method [1], the Hue-Intensity-Saturation (HIS) transform method [2], the Principal Component Analysis (PCA) method [3] and the wavelet transform (WT) method [4-5]. The HPF method improves the spatial details, but produces serious noise. The HIS transform method directly replaces the component *I* of the multi-spectral image with the high-resolution panchromatic image, and it improves the spatial details of the multi-spectral image, but produces serious spectral information distortion because the component *I* contains spectral information. The PCA method replaces the first principle component of the multi-spectral image with the panchromatic image, and it improves the spatial details, but also seriously distorts spectral information. The WT method is to replace high frequency coefficients of the multi-spectral image with corresponding components of the panchromatic image in the

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 481 – 488, 2006.

transform domain. If the decomposition level is too small, the fusion image preserves spectral characteristics of the multi-spectral image, but fails to improve the spatial details well because the discarded low frequency coefficient of the panchromatic image still contains many spatial details. When the level is increased, the performance capacity of the spatial details gradually increased in the fusion image, but the spectral information is not preserved well because the low frequency coefficient is decomposed time after time, and the mosaic phenomenon may be produced. To resolve the conflict, the usual method is to find the balance between performance capabilities of the spectral information and the spatial details with the adjustment of the wavelet decomposition level.

The fusion method based on PCA, HPF and UDWT (undecimated discrete wavelet transform) is proposed through the performance analysis of the classical image fusion methods. The performances of the new method are tested by merging the SPOT panchromatic image and the TM multi-spectral image, and experimental results verify the validity of the method. With the same wavelet decomposition level, the new method has the advantage of preserving more spatial details and distorting less spectral information in comparison with the traditional wavelet transform method.

2 Principal Component Analysis

Principal Component Analysis (PCA) is one of the linear mapping techniques. To fix notations, consider *n* wave bands multi-spectral images as the vector *X*

$$X = [x_1, x_2, x_3, \cdots, x_n]^T \ . \tag{1}$$

The variance between different wave bands is denoted as

$$\delta_{ij}^2 = E[(x_i - m_i)(x_j - m_j)], i, j = 1,2,3,\cdots n \ . \tag{2}$$

where x_i and x_j are the means of the wave band *i* and *j* images. The symmetric covariance matrix is Σ

$$\Sigma = \begin{bmatrix} \delta_{1,1} & \delta_{1,2} & \cdots & \delta_{1,n} \\ \delta_{2,1} & \delta_{2,2} & \cdots & \delta_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{n,1} & \delta_{n,2} & \cdots & \delta_{n,n} \end{bmatrix} . \tag{3}$$

The covariance matrix Σ is then diagonalized, and the eigenvectors φ_r $(r = 1,2,\cdots,n)$ are calculated according to the corresponding eigenvalues from high to low. The eigenvectors vector is given by

$$\phi_n = [\varphi_1, \varphi_2, \varphi_3, \cdots, \varphi_n]^T \ . \tag{4}$$

The *n* wave bands multi-spectral images are mapped onto the eigenvector

$$Y = \phi_n \cdot X \ . \tag{5}$$

In the PCA method, the multi-spectral images are transformed with PCA, and the principle components $y_i\,(i=1,2,\cdots,n)$ are obtained. The panchromatic image is matched by the first principle component with the histogram matched method, and the first principle component y_1 is replaced with the matched panchromatic image. The fusion image is obtained when the new first principle component and the other principle components are transformed with the inverse PCA transform. The PCA method improves the spatial details of the multi-spectral images, but it produces serious spectral information distortion because the first component of the multi-spectral images contains much spectral information.

3 Undecimated Discrete Wavelet Transform

With the ability of multi-solution analysis and multi-resolution image decomposition, the wavelet transform has been employed for remote sensing image fusion. According to the discrete wavelet transform (DWT) method [4-5], the high frequency coefficients of the multi-spectral image are replaced with those of the panchromatic image in the wavelet transform domain. The fused image is synthesized by the inverse discrete wavelet transform (IDWT). The multi-resolution analysis of the DWT does not preserve the translation invariance because of subsampling following each filtering stage. The wavelet coefficient of an image discontinuity could disappear arbitrarily. To preserve the translation invariance, the undecimated discrete wavelet transform (UDWT) method has been introduced [6]. The downsampling operation is suppressed, and the filters of the level j are acquired by 2^j upsampling the DWT filters

$$h_k^{[j]} = h_k \uparrow 2^j = \begin{cases} h_{k/2^j}, & k = 2^j m, \ \ if \quad m \in Z \\ 0, & else \end{cases}$$
$$g_k^{[j]} = g_k \uparrow 2^j = \begin{cases} g_{k/2^j}, & k = 2^j m, \ \ if \quad m \in Z \\ 0, & else \end{cases} \quad . \tag{6}$$

The frequency response of Eq.(6) will be $H(2^j w)$ and $G(2^j w)$ respectively. The coefficients of the level j+1 obtained from the level j are the following

$$\begin{aligned} A_{j+1}(m,n) &= \sum_k \sum_l h_k^{[j]} h_l^{[j]} A_j(m+k,n+l) \\ W_{j+1}^{LH}(m,n) &= \sum_k \sum_l g_k^{[j]} h_l^{[j]} A_j(m+k,n+l) \\ W_{j+1}^{HL}(m,n) &= \sum_k \sum_l h_k^{[j]} g_l^{[j]} A_j(m+k,n+l) \\ W_{j+1}^{HH}(m,n) &= \sum_k \sum_l g_k^{[j]} g_l^{[j]} A_j(m+k,n+l) \end{aligned} \ , \tag{7}$$

where (m,n) stands for the pixel position, A_j is the approximation of the original image at the scale 2^j, and three high frequency components W_j^{LH}, W_j^{HL} and W_j^{HH} corresponding to horizontal, vertical and diagonal spatial details. The scheme of the decimated discrete wavelet coefficient decomposition and reconstruction is depicted in Fig. 1(a), and the scheme of the undecimated discrete wavelet transform is shown in Fig. 1(b).

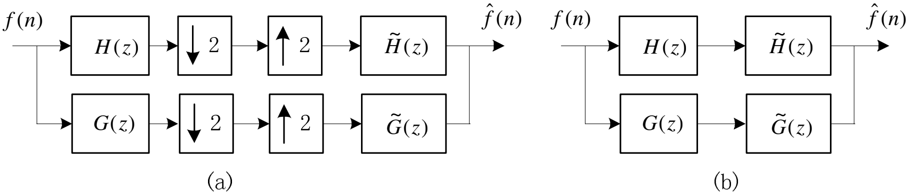


Fig. 1. Discrete wavelet decomposition and reconstruction. (a) Decimated, (b) Undecimated

4 The Fusion Method Based on PCA and UDWT

To improve the performance of the spatial details when preserving the spectral information, the new fusion method makes use of PCA, HPF and UDWT. The panchromatic image is first processed by HPF, and the fused image preserves spectral information and spatial details well when the wavelet decomposition level is small.

The whole processing program of the realization is as follows:

Step1. The multi-spectral images are transformed with PCA, and the panchromatic image is processed with HPF.

Step2. The low frequency part of the panchromatic image is matched by the first principle component of the multi-spectral image with the histogram matched method.

Step3. The matched low frequency part of the panchromatic image and the first principle component are both transformed with the undecimated discrete wavelet transform. Two sets of undecimated wavelet coefficients are obtained, including approximation (*LL*) and detail (*HL*, *LH* and *HH*) components of the original data. The first principle component of the multi-spectral image is reconstructed through the fusion process of the wavelet domain and inverse UDWT. The fusion rule in the transform domain is introduced as follow:

(1) At the level 2^j, the low frequency approximate coefficient used in the fusion process is the *LL* coefficient of the multi-spectral image.

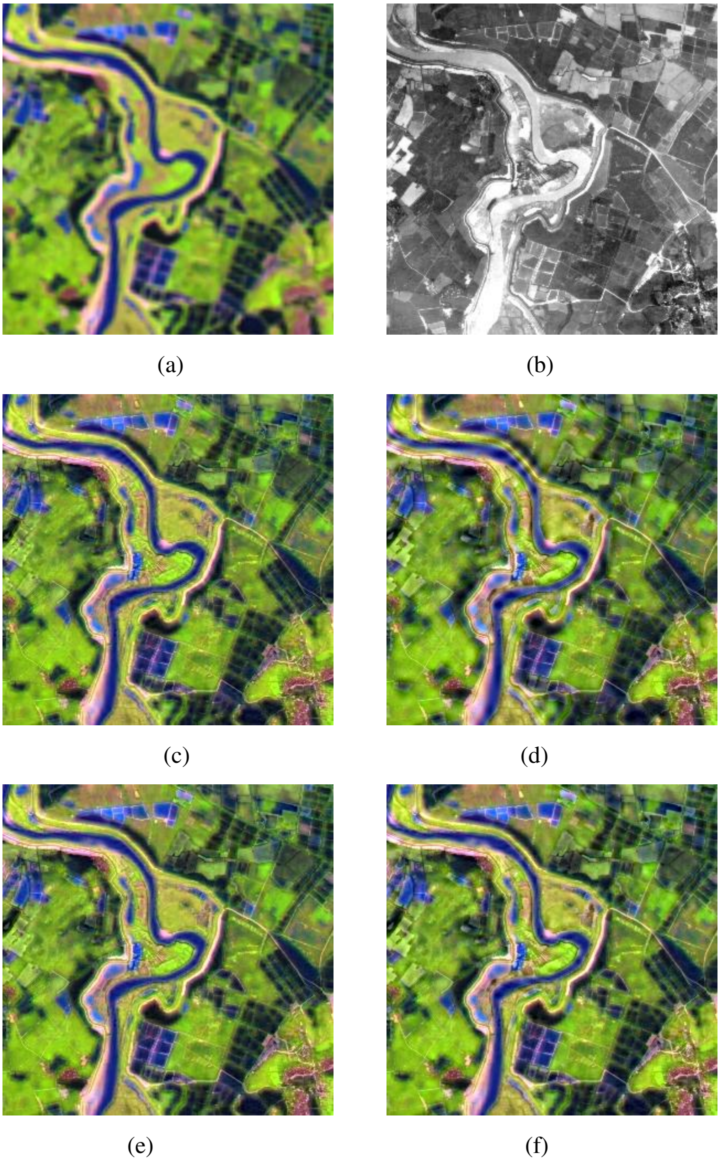
(2) At each level, the high frequency coefficient with the higher gradient value between two sets of detail components is adopted in each direction.

Step4. The high frequency part of the panchromatic image is added to the reconstructed first principle component, a new first principle component of the multi-spectral image is acquired.

Step5. Finally, the new first principle component and the other principle components are transformed with the inverse PCA to obtain the fusion image.

5 Experimental Results and Performance Evaluation

The registration TM multi-spectral image and SPOT panchromatic image are used to verify the validity of the new method. The TM image is shown in Fig. 2(a), and the SPOT panchromatic image is illustrated as Fig. 2(b). The fusion image with the 2-level DWT method is illustrated as Fig. 2(c), and the fusion image with the 3-lelve DWT method is shown in Fig. 2(d). The fusion image using the new method with 2-level UDWT is illustrated as Fig. 2(e), and Fig. 2(f) is the fused image using the new method with 3-level UDWT.



(a) (b)

(c) (d)

(e) (f)

Fig. 2. The original images and fusion images. (a) The TM multi-spectral image. (b) The SPOT panchromatic image. (c) The fusion image with the 2-level DWT method. (d) The fusion image with the 3-level DWT method. (e) The fusion image using the new method with 2-level UDWT. (f) The fused image using the new method with 3-level UDWT.

Generally, the performance evaluation of the image fusion method can be divided into two ways, namely, subjectively qualitative analysis and objectively quantitative evaluation.

5.1 Subjectively Qualitative Analysis

Subjectively qualitative analysis mainly includes two areas:

(1) The visual quality of the fused image, such as spatial resolution, clarity, contrast, sophisticated details, etc.

(2) The spectral fidelity, it indicates the extent of preserving original spectral signal or spectrum characteristics.

Fig. 2(c) is the fusion image with the traditional 2-level DWT method. When the number of decomposition level is increased to 3, the fusion image Fig. 2(d) preserves more spatial details, especially in the left part of the image, but has increased spectral distortion, such as the river region of the image.

Fig. 2(e) is the fusion image using the new method with 2-level UDWT, the spatial details is more sophisticated than those of Fig. 2(c), and their spectral information are similar. Fig. 2(f) is the fused image using the new method with 3-level UDWT, its spatial details is more sophisticated than those of Fig. 2(e), but spectral information distortion begins. Compared Fig. 2(f) with Fig. 2(d), the new method preserves spectral information and spatial details well, i.e. the new method provides the better fusion solution than the DWT method with the same decomposition level.

5.2 Objectively Quantitative Evaluation

In the way of subjectively qualitative judgment, different results could be acquired by reason of differences between individual visual and psychological factors, and professional experience of observers will also affect the final conclusion. Therefore, it is necessary to define a series of quantitative evaluation parameters of the visual quality and spectral fidelity. The current quantitative parameters mainly include mean, standard deviation, average error, entropy, entropy difference, average gradient value, deviation index, correlation coefficient, etc. The information entropy, average gradient and deviation index are used to measure fusion results of different methods.

5.2.1 Entropy

Entropy is an important index to measure the information deposited in images. According to the principle of Shannon information theory, the entropy of the 8-bit image can be defined as

$$H(x) = -\sum_{i=0}^{255} P_i \log_2 P_i \quad , \tag{8}$$

where p_i is the probability of the gray i in the image.

5.2.2 Average Gradient

Average gradient is sensitive to minor details of the image. It can be used to assess the ambiguous extent of the image, and is calculated as

$$\Delta\overline{g} = \frac{1}{M \cdot N}\sum_{x=1}^{M}\sum_{y=1}^{N}\sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2} \ . \tag{9}$$

Generally, the greater the average gradient, the clearer the image.

5.2.3 Deviation Index

Deviation index is introduced to measure the deviation extent between the fused image and the original multi-spectral image. It is defined as follows:

$$DI = \frac{1}{M \cdot N}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{|FUS(i,j) - MUL(i,j)|}{MUL(i,j)} \ , \tag{10}$$

where *FUS* is the fused image, *MUL* is the original multi-spectral image. Generally, the greater the deviation index, the more serious the spectral distortion.

Table 1 shows the quantitative evaluation of the fused images with three parameters. The Deviation index is acquired by calculating the deviation between the intensity component *I* of the original multi-spectral image and the intensity component *I* of the fused image.

Table 1. The statistical comparison of the fusion results

Image	Entropy	Average gradient	Deviation index
The Panchromatic image	7.6764	22.2506	
The Multi-spectral image	5.9120	9.4943	
The fusion image with the DWT method(2 level)	7.5342	19.8748	0.1104
The fusion image with the DWT method(3 level)	7.5826	20.3007	0.1717
The fusion image with the new method(2 level)	7.4077	20.2581	0.1087
The fusion image with the new method(3 level)	7.5314	20.8206	0.1376

The entropies of four fusion images are all increased as compared with the original multi-spectral image. Compared with the fusion image using 2-level DWT method, the average gradient of the fusion image using 3-level DWT method becomes greater, and the same is the deviation index. It indicates that spatial details are enhanced, but the distortion of spectral information is exacerbated. The new method with 2-level UDWT is superior to the 2-level DWT method in the average gradient, and is similar in the deviation index. Compared with the fusion image using the new method with 2-level UDWT, the average gradient of the fusion image using the new method with 3-level UDWT is greater, and the distortion of spectral information is increased. But the new method with 3-level UDWT is superior to the 3-level DWT method in the average gradient and the deviation index of the fused image. It is obvious that the conclusion of quantitative data evaluation consists with the above conclusion of subjectively qualitative analysis.

Synthesized the conclusions of subjectively qualitative analysis and objectively quantitative evaluation, it is concluded that the new method not only distinctly improves the spatial details but also preserves more spectral information of the multi-spectral image. With the same wavelet transform level, the fusion image using the proposed method has more sophisticated spatial details, and distorts less spectral information compared with the fusion image using the DWT method.

6 Conclusion

A remote sensing image fusion method based on PCA, HPF and undecimated discrete wavelet transform is presented. The performances of the proposed method are tested by merging the SPOT panchromatic image and the TM multi-spectral image. Both subjectively qualitative analysis and objectively quantitative evaluation verify the validity of the new method. The multi-spectral image contains abundant spectral information, but lacks in spatial details owing to the lower resolution. The panchromatic image is rich in details. The new method can improve the spatial details while preserving the spectral information of the multi-spectral image. Compared with the traditional discrete wavelet transform method of the same wavelet transform level, the new method has the advantage of preserving more spatial details and spectral information.

References

1. Shettigara VK. A generalized component substitution technique for spatial enhancement of multispectral images using a higher resolution data set. Photogrammetric Engineering and Remote Sensing 1992; 58(5):561-567.
2. Carper WJ, Lillesand TM, Kiefer RW. The use of intensity-hue-saturation transformations for merging SPOT panchromatic and multispectral image data. Photogrammetric Enginering and Remote Sensing 1990; 56(4):459-467.
3. Chavez PS, Sides SC, Anderson JA. Comparison of three different methods to merge multi-resolution and multi-spectral data: Landsat TM and SPOT panchromatic. Photogrammetric Engineering and Remote Sensing 1991; 57(3):295-303.
4. Li H, Manjunath BS, Mitra SK. Multisensor image fusion using the wavelet transform. Graphical Models and Image Processing 1995; 27(3):235-244.
5. Nunez J, Otazu X, Fors O, Prades A, Pala V, Arbiol R. Multiresolution-Based Image Fusion with Additive Wavelet Decomposition. IEEE Trans on Geosciences and Remote Sensing 1999; 37(3):1204-1211.
6. Bruno A, Luciano A, Stefano B, Andrea G. Context-Driven Fusion of High Spatial and Spectral Resolution Images Based on Oversampled Multiresolution Analysis. IEEE Trans on Geosciences and Remote Sensing 2002; 40(10):2300-2312.

Speech Recognition with Multi-modal Features Based on Neural Networks

Myung Won Kim[1], Joung Woo Ryu[2], and Eun Ju Kim[1]

[1] School of Computing, Soongsil University, 511, Sangdo-Dong, Dongjak-Gu, Seoul, Korea
mkim@comp.ssu.ac.kr, blue7786@naver.com
[2] Intelligent Robot Research Division Electronics and Telecommunications Research Institute
161 Gajeong-dong, Yuseong-gu, Daejeon, Korea
ryu0914@etri.re.kr

Abstract. Recent researches have been focusing on fusion of audio and visual features for reliable speech recognition in noisy environments. In this paper, we propose a neural network based model of robust speech recognition by integrating audio, visual, and contextual information. Bimodal Neural Network (BMNN) is a multi-layer perceptron of 4 layers, which combines audio and visual features of speech to compensate loss of audio information caused by noise. In order to improve the accuracy of speech recognition in noisy environments, we also propose a post-processing based on contextual information which are sequential patterns of words spoken by a user. Our experimental results show that our model outperforms any single mode models. Particularly, when we use the contextual information, we can obtain over 90% recognition accuracy even in noisy environments, which is a significant improvement compared with the state of art in speech recognition.

Keywords: speech recognition, neural network, post-processing, contextual information, sequential pattern.

1 Introduction

As the technology of mobile devices advances and such devices come into wide use, speech becomes one of important human computer interfaces (HCI). Recently, a study of multi-modal speech recognition is in progress to realize easier and more precise human computer interfaces. Particularly, the bimodal speech recognition has been studied for high recognition rate at environments with background noise. In the bimodal speech recognition if the audio signal is of low quality or ambiguous, visual information, i.e. lip-movements can contribute to the recognition process as well.

In the bimodal speech recognition the most important issues are how well we extract the visual information, as supplementary to the audio signal, and how efficiently we merge these different modes of information. We investigate the second issue which is called the 'information fusion' problem.

The existing fusion methods are divided into feature fusion and decision fusion depending on the point of time that different sources of information are fusioned [1]. The feature fusion is a method which fuses features extracted from different sources

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 489 – 498, 2006.

of information to produce the recognition results, while the decision fusion is a method which combines the recognition results of various independent recognizers to produce the final result. The HMM (Hidden Markov Model) and the neural networks are models generally used to implement these fusion methods.

[2] proposed a feature fusion method using the HMM. Feature fusion has the synchronization problem because the sampling rates of audio and visual information are different. The low-pass interpolation method is used to extract samples to solve the synchronization problem, and a new feature was created from the 25msec window where 10msec is overlapped. However, it is difficult to decide the number of states and the number of Gaussian mixtures which correspond to the learning variables that show a sensitive response to the fusion method using the HMM. Moreover, it is especially difficult to apply the generally used CDMM(Continuous Density Hidden Markov Model) because of restriction that its input features satisfy probabilistic independence [3][4].

The TDNN (Time-Delay Neural Network) is a neural network which can recognize phonemes. It has two important properties. 1) Using a 3 layer arrangement of simple computing units, it can represent arbitrary nonlinear decision surfaces. The TDNN learns these decision surfaces automatically using error back-propagation. 2) The time-delay arrangement enables the network to discover acoustic-phonetic features and the temporal relationships between them independent of position in time and hence not blurred by temporal shifts in the input[5]. The MS-TDNN(Multi-State TDNN) is an expanded model of TDNN to recognize continuous words by adding the DTW(Dynamic Time Wrapping) layer[6][7]. Using the MS-TDNN, the bimodal MS-TDNN which integrates audio and visual features was proposed in [8].

The bimodal MS-TDNN is constructed through the two-level learning process. In the first learning process the preprocessed acoustic and visual data are fed into two front-end TDNNs, respectively. Each TDNN consists of an input layer, one hidden layer and the phone-state layer. Back-propagation was applied to train the networks in bootstrapping phase, to fit phoneme targets. Above the two phone-state layers, the DTW algorithm is applied to find the optimal path of phone-hypotheses for the word models. In the word layer the activation of the phone-state units along the optimal paths are accumulated. The highest score of the word units represents the recognized word. In the second learning process the networks are trained to fit word targets. The error derivatives are back-propagated from the word units through the best path in the DTW layer down to the front-end TDNNs, ensuring that the network is optimized for the actual evaluation task, which is word and not phoneme recognition.

The DTW algorithm is required to solve the time axis variation problem because the bimodal MS-TDNN needs to recognize words from phonemes. Therefore, the MS-TDNN is complex in structure and it still has the problem that its performance is sensitive to noise.

In this paper, we propose the BMNN(Bimodal Neural Network), a neural network model for isolated word recognition, which can efficiently combines diverse sources of information. To improve speech recognition accuracy in noisy environments we also propose the post-processing method using contextual information such as sequential patterns of the words spoken by the user.

This paper is organized as follows. Section 2 describes the methods for extraction of audio features from speech signal and visual features from lip movement images.

Section 3 explains our proposed bimodal neural network model, and Section 4 describes the post-processing method using contextual information to improve the speech recognition accuracy. Section 5 discusses the experiments with the proposed method, and finally Section 6 concludes the paper.

2 Audio and Visual Feature Extraction

In this paper we adopt the existing feature extraction methods, the ZCPA(Zero Crossing with Peak Amplitude)[9] method for audio features, and the PCA(Principle Component Analysis) method for visual features. In the following we describe these methods in detail.

2.1 Audio Feature Extraction

The ZCPA models the auditory system to the auditory nerve, which is composed of the cochlear filter bank and a nonlinear transformer connected to the output of each cochlear filter bank. The cochlear filter bank is a modeling of the basilar membrane just like the general auditory model, where the nonlinear transform block is the modeling of the stimulating process of the nerve cell through a mechanical vibration of the basilar membrane, and is connected in series with the linear filters.

The ZCPA is composed of a 16 channel filter bank block, a zero-crossing detection block, a nonlinear transform block, and a feature extraction block. The filter bank is composed of a FIR filter which has the powers-of-two coefficients, and made frequency calculations of high precision possible using bisections recursively. The bisecting method and the binary search method are used in the nonlinear transform block which increases the calculation speed and the memory size. Lastly, the feature vector was extracted for the feature extraction method by accumulating the maximum value which is non-linearized to the corresponding frequency band in the size of the frame of each filter bank.

2.2 Visual Feature Extraction

The most widely used method for visual feature extraction is the PCA, which is a transformation of data based on statistical analysis. The PCA reduces the visual input dimension through statistical analysis, and has the property that it preserves important information even with the reduced dimensions. We extract a basis for representing visual features of an image through the PCA. The given 16x16 sized image of speaker's lips can be represented as a linear combination of those basis as in Fig. 1. Here, (c_1, c_2 , ..., c_n) is a feature vector representing the image of a speaker's lips.

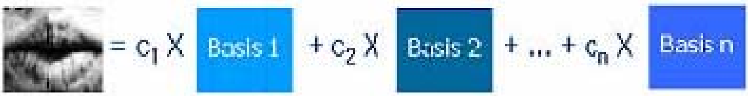


Fig. 1. Lip image representation

When an image stream consists of M frames, M n-dimensional vectors are calculated and represent the visual features of lips movement. However, the extracted features are different between speakers, so it would be better to represent the lips movement by the difference between the feature vector and the average of the feature vectors of the images of M frames as described in equation (1).

$$\begin{aligned} \bar{u} &= \frac{1}{M}\sum_{i=1}^{M} u_i \\ v_k &= u_k - \bar{u} \quad , k = 1, 2, \ldots, M \end{aligned} \tag{1}$$

where u_k is the feature vector for the k-th frame, and v_k is the extracted feature vector for the k-th frame. In this paper we set the dimension of the feature vector (n) and the number of frames (M) to 16 and 64, respectively. In addition, we use the interpolation method to create the feature vectors for 64 frames because the number of input vectors need to be fixed according to the structure of the recognizer.

3 BMNN (Bimodal Neural Network)

This paper proposes the bimodal speech recognition model that is robust in noisy environments using neural network. The proposed BMNN structure is as shown in Fig. 2.

The BMNN consists of 4 layers (input layer, hidden layer, combined layer, output layer) and is designed as a feed-forward network with the error back-propagation algorithm as the learning algorithm. Since we deal with isolated word recognition, an overlap zone structure is used which shows high performance for isolated word recognition [10]. The third layer combines audio and visual features of speech to compensate loss of audio information caused by noise.

When the connection structure of the model and the number of frames of each layer is observed, the nodes of the upper layer frame and the corresponding nodes of every frame included in the window are fully connected, and the combined layer is also fully connected to the output layer because no windows are used. Therefore, the number of frames for each layer is automatically determined by equation (2), when the number of lower layer frames, the size of the window, and the size of the overlap zone is determined. This paper set the value of equation (2) to be a constant for the size of the overlap zone, and the number of feature frames of each layer is set so that the number of feature frames of the lower layer is reduced in half each time to it through the experiment. A structure like this can be more efficient compared with the model in [11] because the size of the model and the number of connections reduce.

$$HF = \frac{LF - O}{W - O} \tag{2}$$

where HF and LF represent the number of frames of the upper layer and the number of frames of the lower layer, respectively, and W is the window size, and O is the overlap zone size.

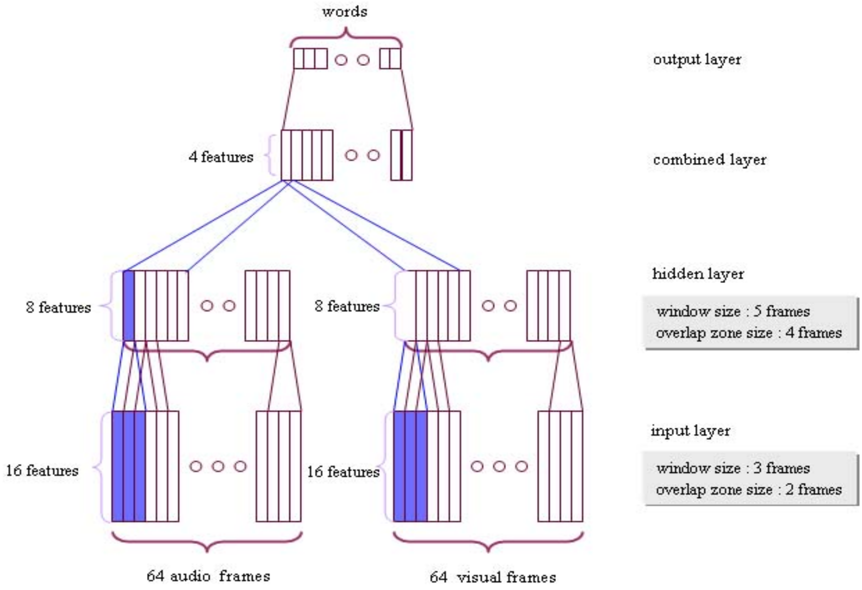


Fig. 2. BMNN architecture

The BMNN recognizes isolated words so we do not have the problem of time axis variation. Therefore, it has the advantage that the learning method and the structure is simpler compared with the bimodal MS-TDNN of phoneme units. We take the advantage of neural network that it allows more efficient fusion of heterogeneous information than the HMM. However, the BMNN with the feature fusion method has the problem that speech and visual information must be synchronized properly. For that reason, the image captured by a camera is stored together the system tick into the visual buffer, and simultaneously the speech signal is input from the microphone, and then the input speech signal is segmented to an isolated word using the endpoint detecting algorithm. At this moment, the tick which indicates the same time as the endpoint detecting time is calculated, and the image which is input at the identical time (starting time ~ finishing time) from the image buffer, is read in from the buffer. Through this process, the extracted images and speech signals are synchronized by extracting feature vectors using the feature extraction method described in section 2.

4 Post-processing of Speech Using Contextual Information

The need of speech recognition that is robust in noisy environments is rising due to the wild use of mobile devices. Therefore, we propose a post-processing method of speech to improve the recognition accuracy using contextual information such as sequential patterns of words spoken by the user.

4.1 Context Recognizer

The context recognizer which recognizes sequential patterns of commands is a multilayer perceptron with 3 layers. The context recognizer predicts the current command from a sequence of preceding commands. Its input layer represents a sequence of preceding commands while the output layer represents the current command. In this research we adopt local coding in representing data both in the input layer and the output layer, in which each command is represented by a single node. If we take the total number of commands in use and the length of sequences of preceding commands to be *n* and *m*, respectively, then we have *m* blocks of *n* nodes in the input layer, while we have *n* nodes in the output layer. A sequence of commands is mapped into geographical positions of input nodes. For example, the first command in a sequence into the left most block of input nodes and the second command into the next left block of nodes and so on.

This structure of neural network is used to capture useful sequential patterns of commands that a user utters. Once the model is trained using the training data, the model learns sequential patterns of commands and can predict the current commands given a sequence of preceding commands.

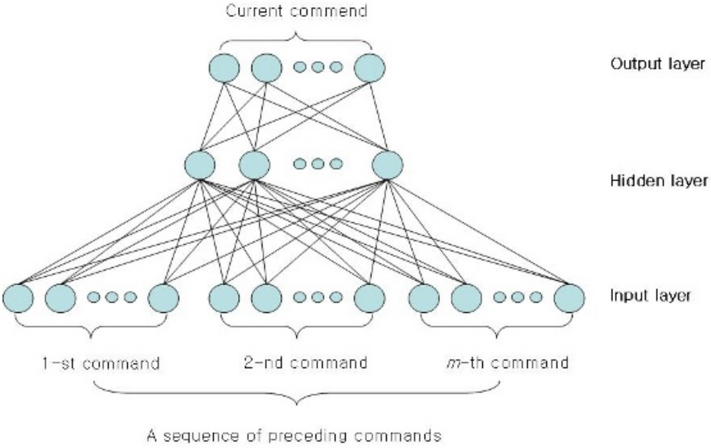


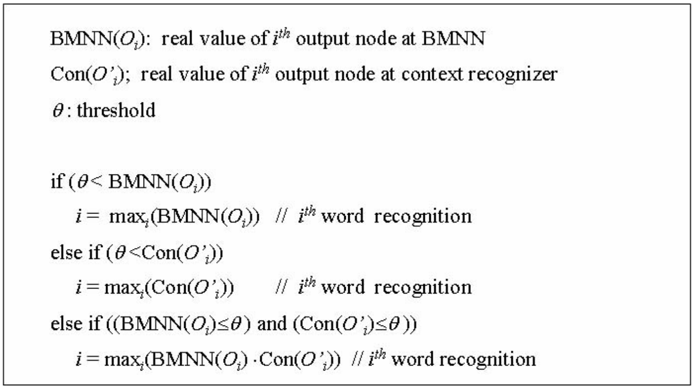
Fig. 3. Context recognizer architecture

4.2 Post-processing Using Word Sequence Patterns

The structure of speech recognition with the post-processing method is shown in Fig. 4. The final recognition result is given by combining the output of the BMNN recognizer and the output of the context recognizer.

To efficiently combine the results of the two recognizers, a sequential combination method is used as shown in Fig. 4. In the method we take the word of the maximum output value of the BMNN if any output value of the BMNN is greater than the given threshold(θ). Else, we take the word of the maximum output value of the context recognizer if any output value of the recognizer is greater than the threshold.

Otherwise, we assume that none of the recognition result is reliable and the output values of the two recognizers are multiplied and the word of the largest value is selected to be the final recognition result. The threshold is given by the user and it means the lower margin of the degree that the user can rely on the recognition result.



```
$BMNN(O_i)$: real value of $i^{th}$ output node at BMNN
$Con(O'_i)$; real value of $i^{th}$ output node at context recognizer
$\theta$: threshold

if ($\theta <$ BMNN($O_i$))
    $i$ = max$_i$(BMNN($O_i$))   // $i^{th}$ word recognition
else if ($\theta <$Con($O'_i$))
    $i$ = max$_i$(Con($O'_i$))        // $i^{th}$ word recognition
else if ((BMNN($O_i$)$\leq\theta$) and (Con($O'_i$)$\leq\theta$))
    $i$ = max$_i$(BMNN($O_i$) $\cdot$Con($O'_i$))  // $i^{th}$ word recognition
```

Fig. 4. Sequential combination algorithm

5 Experiments

The speech data used in our experiments is speaker dependent data produced by the ETRI(Electronics and Telecommunications Research Institute). The speech data constitutes of 35 isolated Korean words spoken 27 times. The words are commands which can be used in a mobile device. The noisy data was generated by artificially adding Gaussian noises (20db, 10db, 5db) to simulate speech signal in noisy environments.

The model structure of the BMNN used in this experiment set the number of input frames to 64 (10ms per frame) for isolated word recognition and extracted 16 features from each frame. The window size of the input layer is set to 3 frames of 30ms which is sufficient to represent a phoneme, while the overlap zone size is set to 2 frames. The window size of the hidden layer is set to 5 frames while the overlap zone size is set to 4 frames. Therefore, the number of frames of the hidden layer is 62 and that of the combined layer is 58 on the basis of equation (2).

To verify the efficiency of the BMNN in noisy environments, performance is compared with single mode recognizers using either audio features or visual features only. A single mode recognizer (the speech recognizer or the image recognizer) can be achieved simply by the BMNN with ‘0’ as input for audio features or visual features.

5.1 Speaker Dependent Recognition Without Post-processing

For the speaker dependent recognition we made 350 training data of 35 isolated Korean words spoken 10 times and 595 test data spoken 17 times.

When the SNR is 30db, there is no significant difference between the speech recognizer’s performance (94.43%) and the BMNN’s performance (95.49%), but when the noise level increases the recognition accuracy decreases by 13.16% for the speech recognizer, while the performance decrease of the BMNN is 8.2%, which is about 3.96% lower. We can notice that the visual features contribute more significantly when there is more noise in speech.

Table 1. Performance comparison without post-processing

	SNR			
	30db	20db	10db	5db
Speech	94.48	89.24	72.54	54.99
Visual	51.08	51.08	51.08	51.08
BMNN	95.49	94.13	85.81	70.89

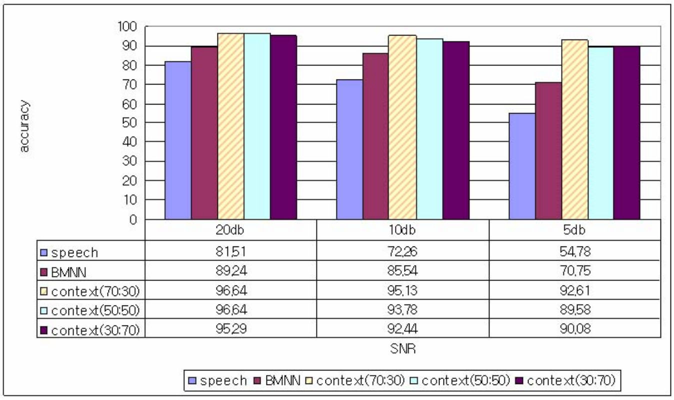
5.2 Experiment with Post-processing

In this experiment we define sequential command patterns which are spoken by the user for a mobile device as the contextual information, and we demonstrate the efficiency of the post-processing method in noisy environments.

First of all, it is assumed that sequential patterns among commands which are spoken by the user and we create a context recognizer modeled by a multi-layer perceptron which learn such sequential patterns. In our experiment, we create three context recognizers for different regularities 70%, 50% and 30% of command sequential patterns. For example, for a sequential patterns of regularity 70%, 'start browser, favorite sites, item 5' is uttered as a sequence of commands, and the probability of the 'select' command being followed is 70%, while the probability of any other command followed is 30%. The reason that we consider different regularities is to find out how much the regularity of command sequential patterns affects the performance of post-processing of speech recognition.

For the context recognizer we set the number of preceding commands to 3. The multi-layer perceptron as a context recognizer consists of three layers and 105 nodes for the input layer, 52 nodes for the hidden layer and 35 nodes for the output layer. We have 105 input nodes since there are 35 commands to recognize and the number of the preceding commands is 3. The number of hidden nodes was obtained experimentally.

Fig. 5 compares the performances of the single mode speech recognizer, BMNN and BMNN with post-processing. We can see that the average performance of the speech recognizer is 69.51% for the noise levels of 20 db, 10 db, and 5 db, and that of the BMNN is 81.84%, while the BMNN with the post-processing shows the average accuracy of 93.57%. Also, the average reduction of performance was examined as the noise level increases. The speech recognizer shows the average performance decrease of 13.36% and the BMNN shows 9.24%, while for BMNN with post-processing the average performance decrease is 2.72%. We can see that the BMNN with post-processing is very little affected by noise. This paper clearly demonstrates the possibility of exploiting contextual information such as sequential patterns of commands of the user for improved speech recognition accuracy, particularly in noisy environments.



	20db	10db	5db
▨ speech	81.51	72.26	54.78
■ BMNN	89.24	85.54	70.75
□ context(70:30)	96.64	95.13	92.61
□ context(50:50)	96.64	93.78	89.58
■ context(30:70)	95.29	92.44	90.08

Fig. 5. Experimental result with post-processing

6 Conclusion

This paper has proposed the BMNN, which can efficiently fusion the audio and visual information for robust speech recognition in noisy environments. BMNN is a multilayer perceptron of 4 layers, each of which performs a certain level of abstraction of input features. In the BMNN the third layer combines audio and visual features of speech to compensate loss of audio information caused by noise. In order to improve the accuracy of speech recognition in noisy environments, we also proposed a post-processing based on contextual information such as sequential patterns of words spoken by a user. Our experimental results show that our model outperforms any single mode models. Particularly, when we use the contextual information, we can obtain over 90% recognition accuracy even in noisy environments, which is a significant improvement compared with the state of art in speech recognition. Our research demonstrates that other sources of information need to be integrated to improve the accuracy of speech recognition particularly in noisy environments.

For future research, we need to investigate diverse sources of contextual information such as the topics or situation of speech that can be used to improve speech recognition. We will also investigate a more robust method for integrating different sources of information in speech recognition.

Acknowledgments. This works was supported by the Research Program of Soongsil University.

References

1. Chibelushi, C.C., Deravi, F., Mason, J.S.D.: A Review of Speech-Based Bimodal Recognition. IEEE Transactions on Multimedia, Vol. 4, No. 1 (2002) 23-37
2. Kaynak, M.N., Qi Zhi, Cheok, A.D., Sengupta, K., Ko Chi Chung: Audio-visual modeling for bimodal speech recognition. Proceedings of the IEEE Systems, Man, and Cybernetics Conference, Vol. 1 (2001) 181-186

3. Gemello, R., Albesano, D., Mana, F., Moisa, L.: Multi-source neural networks for speech recognition: a review of recent results. Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, Vol.5 (2000) 265-270
4. Xiaozheng Zhang, Merserratt, R.M., Clements, M.: Bimodal fusion in audio-visual speech recognition. International Conference on Image Processing, Vol.1 (2002) 964-967
5. A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. J. Lang: Phoneme Recognition Using Time-Delay Neural Networks. IEEE Trans. on Acoustics, Speech and Signal Processing. Vol.37, No.3 (1989) 328-339
6. Haffiner,P., Waibel, A.: Multi-State Time Delay Neural Networks for Continuous Speech Recognition. In Advances in Neural Information Processing Systems 4, Morgan Kaufmann Publishers (1992)
7. Joe Tebelskis: Speech Recognition using Neural Networks. CMU-CS-95-142, School of Computer Science Carnegie Mellon University Pittsburgh (1995)
8. C. Bregler, S. Manke, H. Hild, A. Waibel: Bimodal sensor integration on the example of "speech-reading". Proc. of IEEE Int. Conf. on Neural Networks, San Francisco (1993)
9. Doh-Suk Kim, Soo-Young Lee, Rhee M. Kil: Auditory Processing of Speech Signals for Robust Speech Recognition in Real-World Noisy Environments. IEEE Trans. on Speech and Audio Processing, Vol.7, No.1 (1999) 55-69
10. Mary Jo Creaney-Stockton, Beng., MSc.: Isolated Word Recognition Using Reduced Connectivity Neural Networks With Non-Linear Time Alignment Methods. Dept. of Electrical and Electronic Engineering Univ. of Newcastle-Upon-Tyne (1996)
11. Sang Won Lee, In Jung Park: A Study on Recognition of the Isolated Digits Using Integrated Processing of Speech-Image Information in Noisy Environments. Journal of the Institute of Electronics Engineers of Korea, Vol.38-CI, No.3 (2001) 61-67

Speech Feature Extraction Based on Wavelet Modulation Scale for Robust Speech Recognition

Xin Ma[1], Weidong Zhou[1], Fang Ju[1], and Qi Jiang[2]

[1] College of Information Science and Engineering, Shandong University
Jinan, Shandong, 250100, P.R. China
{max, wdzhou, jufang}@sdu.edu.cn
[2] College of Control Science and Engineering, Shandong University
Jinan, Shandong, 250100, P.R. China
jiangqi@sdu.edu.cn

Abstract. An analysis based on wavelet modulation scales feature extraction is proposed. Considering human auditory perception and varieties of disturbances, instead of the frequency differences, wavelet modulation scales are adopted to reflect the dynamic features of speech in ASR. Experiments for the Chinese digit-string recognition show extracting the wavelet modulation scales as the dynamic features have good performance both in additional noises and convolutional noises environment.

Keywords: Feature extraction, Wavelet analysis, Modulation scales.

1 Introduction

Automatic recognition of speech (ASR) has good performance in clean environment, but when speech signal is distorted by noise, the performance of ASR will degrade. Usually the environmental noises are additional and convolutional noises. To eliminate the effects of noises, some methods, such as spectrum subtraction, noise compensation are often introduced and they can effectively suppress the noisy disturbance. But when the condition of environment changes, the results of recognition will become worse.

Noise can be suppressed by improving the robustness of features. As is well known in automatic speech recognition based MFCC features [2], difference and acceleration coefficients are often adopted as auxiliary features to improve the robustness against the noises [3], and they are good dynamic features of speech. Similarly, modulation spectrum is another feature that can well reflect the dynamic feature of speech. In area of modulation spectra, the components that irrelevant to the recognition can be easily separated from the speech features [4].

Usually we can get modulation spectrum by Fourier transform, however, some other studies [5] suggest human perception for modulation accords to a constant-Q property, directly applying Fourier transform can only get uniform distribution in frequency area. To mimic this constant-Q property of human perception, in this paper, we adopt the wavelet transform to get modulation scales as speech features and use

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 499 – 505, 2006.

them in ASR, and use normalizing technique to improve the robustness of speech features against noises. Experiments for the Chinese digit-string recognition prove these approaches have good effects for recognition rate under noisy environments.

The paper is organized as follows: in section 2, first, the theory of modulation spectrum and wavelet modulation scale features are described, then the normalizing process are presented. Experiments and analysis of results are shown in section 3, finally the conclusions are given in Section 4.

2 Modulation Spectrum and Wavelet Modulation Scale

2.1 Theory of Modulation Spectrum

The actual modulation transform is based on the spectrogram, the spectrogram can be defined as

$$|S_x^{(\gamma)}(t,\omega)=STFT_x^{(\gamma)}(t,\omega)|^2 . \tag{1}$$

It complies with principle of quadratic superposition [6], if a signal can be expressed as $x(t)=c_1x_1(t)+c_2x_2(t)$, the spectrogram of $x(t)$ can be written as

$$T_x(t,f)=|c_1|^2T_{x1}(t,f)+|c_2|^2T_{x2}(t,f)+c_1c_2^*T_{x1,x2}(t,f)+c_1c_2^*T_{x2,x1}(t,f) . \tag{2}$$

From above equation, we can see spectrogram of a signal has distinct interference terms. Modulation spectrum can be calculated from spectrogram as follows:

$$M_x(\omega,\eta)=\int_{-\infty}^{+\infty} S_x(t,\omega)e^{-j\eta t}dt . \tag{3}$$

Where ω and η are the acoustic frequency and modulation frequency respectively. $M_x(\omega,\eta)$ can also be viewed as the two-dimensional transform of the instantaneous autocorrelation function, or the correlation function of a Fourier transform $X(\omega)$ [7], but in $M_x(\omega,\eta)$ there are still interference terms ,which can be attenuated by smoothing process using proper window function. Here we use $M^{SP}(\omega,\eta)$ standing for the smoothed $M(\omega,\eta)$, that is

$$M^{sp}(\omega,\eta)=M_W(\eta,\omega)*_{\omega}M_x(\eta,\omega) . \tag{4}$$

$M^{sp}(\omega,\eta)$ is the result of the convolution of $M_x(\eta,\omega)$ and $M_W(\eta,\omega)$ in ω ,the interference terms can be reduced evidently in smoothed modulation features [8]. This conclusion is the base of modulation spectrum using in robustness improvement. The usually steps are as follows: first we frame the speech signal using short windows, the short-time fourier transform is used to acquire the spectrogram, then the spectrogram is divided into subbands in which the modulation frequency transform is performed. As the most useful components for speech recognition in modulation spectrum is between 2-16Hz [9], we select proper bands of modulation frequencies as the speech features.

2.2 Wavelet Modulation Scales

Considering of the constant-Q property of human perception for modulation, instead of fourier transform, we use the wavelet transform for every subbands and acquire the wavelet modulation scales representation. The detailed calculation of speech signal x(t) is as follows:

$$S_y(t,\omega)=\frac{1}{2\pi}|\int x(u)w^*(u-t)\ e^{-jwu}|^2 . \tag{5}$$

$S_y(t,\omega)$ is the spectrogram of $x(t)$, $w^*(t)$ is short-time window function. Along the time directions of $S_y(t,\omega)$,wavelet transform can be witten as

$$W_x(s,\zeta,\omega)=\frac{1}{s}\int S_x(t,\omega)\psi(\frac{t-\zeta}{s})dt\ \cdot \tag{6}$$

$\psi(t)$ is wavelet function, ζ is translation factor, $W_x\ (s,\zeta,\omega)$ is the wavelet modulation scales representation of $x(t)$.

2.3 Modulation Scales Normalization

If a signal $x(t)$ was corrupted by additive noise $d(t)$ and convolutional noise $h(t)$, the noisy signal can be written as

$$y(t)=[x(t)+d(t)]*h(t) . \tag{7}$$

Here for convenience we let $s(t)=x(t)+d(t)$, then the spectrogram of $s(t)$ can be written as

$$S_y(t,\omega)=S_s(t,\omega)S_h(t,\omega) . \tag{8}$$

$S_s(t,\omega)$ and $S_h(t,\omega)$ are the spectrogram of $s(t)$ and $h(t)$, $S_y(t,\omega)$ is windowed along time scales and transformed by wavelet along time scales, the results are the wavelet scale representations of $y(t)$,

$$W_y(s,\zeta,\omega)=\frac{1}{s}\int S_x(t,\omega)W_L(t-B)\psi(\frac{t-\zeta}{s})dt . \tag{9}$$

$W_L(t)$ is window function used for not only avoiding the spectrum leakage but also smoothing the interference terms which was illustrated in equation (2), so here it is called smoothing window function.

If the frequency characteristic of convolutional noises can be thought as linear and time invariant over the smoothing window, we can get following approximate formula,

$$W_y(s,\zeta,\omega) \approx W_s(s,\zeta,\omega)W_h(\omega) . \tag{10}$$

It can be normalized as

$$W_{y,norm}(s,\zeta,\omega)=\frac{W_y(s,\zeta,\omega)}{\int W_y(s,\zeta,\omega)ds}=\frac{W_s(s,\zeta,\omega)W_h(\omega)}{\int W_s(s,\zeta,\omega)W_h(\omega)ds}=\frac{W_s(s,\zeta,\omega)}{\int W_s(s,\zeta,\omega)ds}=W_{s,norm}(s,\zeta,\omega) \tag{11}$$

In actual applying the formula (9) to calculate the wavelet scales, the scale parameter s and translation factor ζ need to be discretized to s_d and ζ_n separately, we can write the discrete representation

$$W_{y,norm}(s_d,\zeta_n,\omega)=\frac{W_s(s_d,\zeta_n,\omega)}{\sum\limits_{s_d} W_s(s_d,\zeta_n,\omega)} . \tag{12}$$

Recently research about modulation spectrum manifests that the distributions of disturbances and the speech signal are different in the whole scales of modulation spectrum [3]. By select the proper scope of s_d, interference terms made from noises can be attenuated, and formula (12) can be further approximated as

$$W_{y,norm}(s_d,\zeta_n,\omega)=W_{x,norm}(s_d,\zeta_n,\omega) . \tag{13}$$

$W_{x,norm}(s_d,\zeta_n,\omega)$ is normalized modulation scale representation of $x(t)$.

3 Experiments and Analysis of Results

The speech signals was framed into 25ms (400 samples) per frame and windowed by hamming window with 8.75ms frame rate. This can acquire 128Hz sampling rate for modulation frequency. For extracting modulation scales features, the bior1.1 function was used as wavelet function. We use Mel subbands instead of uniform frequencies bands for complying the human perception. After we calculated the $S_x(t,\omega)$, we need transform it to representations of the power spectrum under Mel scales. Here we divided the frequencies into 26 Mel subbands (k=26), every subband was framed and windowed by hamming window. For acquiring enough resolution, the frame should have enough length, here the 1s （128 frames） frame length was used, so every long frame include 128 short frame energy values E_n ($0 \le n < 128$). There are 2 dots overlaps because the length of bior1.1 filters are 2. Eight dyadic scales wavelet transforms was conducted to get the modulation scales vectors, the first two values of which should be discarded, like overlap-save method filtering quoted in [11]. Finally, the modulation scales features were normalized and filtered as quoted in section 2.3 of this paper. According to [1], only the third, forth, and fifth layer were saved as modulation scales parameters. So from every long frame we can acquire 3×128 wavelet scales matrix, every column in the matrix was used as parameters of corresponding short frame.

3.1 Recognition Experiments Under Clean Environments

Firstly, we used test set to perform speech recognition experiment under clean environment (no convolutional and additional noise), and assumed that both the training set and the test set are recorded under same channel conditions. The recognition errors on the test set are shown in Table 1.

Table 1. Recognition rate for clean speech

MFCC	MOD	NORM_MOD
7.92%	7.8%	8.6%

From table1, we can see that the performance of above three methods is similar under clean environment.

3.2 Recognition Experiments in Additive Noises

Noises signal n(k) superposed over the clean speech signals as additive disturbance, were extracted from NoiseX-92 database, signal-noise ratio can be determined as

$$\mathrm{SNR}=10\log\left(\sum_{k}|s(k)|^2/\sum_{k}|n(k)|^2\right). \tag{14}$$

Table 2 shows recognition error rate of three methods for test set speech signal corrupted by white, pink, and babble noises. THE SNR of all test speech is 10dB.

Table 2. Recognition error rate of three methods for additive noisy speech

noise / method	white	Babble	pink
MFCC	28.35	32.44	36.12
Mod	20.24	23.61	25.61
Norm_Mod	22.51	27.68	26.44

From Table 2, we can see that the modulation scales features show better robustness under additive noisy environments. Normalized modulation features have good resistance to color noise.

3.3 Recognition Experiments in Convolutional Noises

The environment for a practical recognizer not only has additive noise but may have convolutional disturbance such as telephone network. For simulating the convolutional distortion to the speech, we use a telephone channel impulse response signal to corrupt the tested speech signal. The telephone channel impulse response signal was obtained from a real telephone channels, and its response feature curve was plotted in figure1.

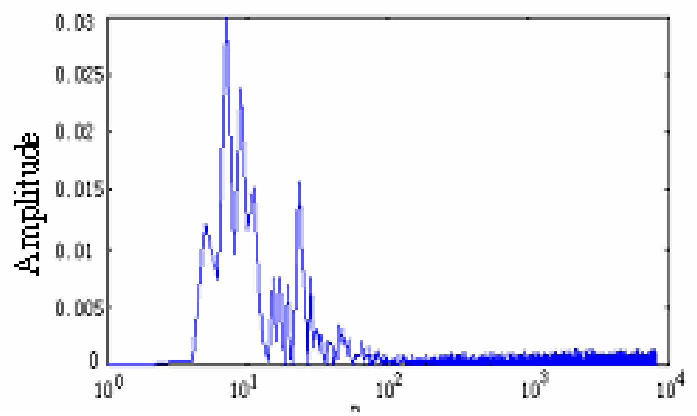


Fig. 1. Channel impulse was obtained from a real telephone channels

The recognition errors on the test set are shown in table 3.

Table 3. Recognition error rate of three methods for convolutional noisy speech

MFCC	MOD	NORM_MOD
21.55%	20.85%	10.4%

The recognition results of MFCC, modulation scales and normalized modulation scales are showed in Table 3. From Table 3, we can see the recognition result of unnormalized wavelet modulation scales features is not very good. However, after normalized, the wavelet modulation scales have good performances under convolutive environments.

4 Conclusion

Modulation spectrum is another way to reflect dynamic features of speech signals. The results of experiments for the Chinese digit-string recognitions show the new method has positive efforts in improving the robustness of speech recognition system. Further research will be done to exploit the modes and extents of its contributions for large vocabulary continuous speech recognition.

References

1. H. Hermansky, Human Speech Perception: Some Lesson From Automatic Speech Recognition (TSD'01, Zelezna Ruda, Czech Republic, Sep, 2001 in TSD'01[DB/OL], Zelezna Ruda)
2. LR.Rabiner and BH.Juang, Fundementals of Speech Recognition (Prentice Hall, Englewood Cliffs, NJ, USA, 1993:194-200)

3. S. Boll. Suppression, of acoustic noise in speech using spectral subtraction, IEEE and Signal Processing, April 1979:113-120
4. H. Hermansky, The Modulation Spectrum in Automatic Recognition of Speech IEEE Workshop on Automatic Speech Recognition and Understanding, 1997:140-147
5. E. R. Kandel, J. H. Schwartz, and T. M. Jessell, ed, Principles of Neural Science Third Edition, Chapter32 Hearing, Elsevier Science Publishing Co., Inc., 1991: 481-498
6. ZhangXian-da, Modern Signal Processing. the Tsinghua University press1995 : 456-457
7. Somsak Sukittanon, Les E. Atlas, Channel Compensation of Modulation Spectral Features, in Proceedings of the 2003 IEEE ISCAS, 2003
8. S. Sukittanon and L. E. Atlas, Modulation Frequency Features for Audio Fingerprinting Proc. of ICASSP'2002, 2002:1173-76
9. Takayuki Arai, Misha Pavel, Hynek Hermansky, and Carlos Avendano, Intelligibility of speech with filtered time trajectories of spectral envelopes Proc. ICSLP-96,Philadelphia, October 1996:2490-2493
10. http://htk.eng.cam.ac.uk/
11. Oppenheim A V, Schafer R W., Digital signal Processing Prentice Hall, Inc.,(1975) 85-86

Fuzzy Controllers Based QoS Routing Algorithm with a Multiclass Scheme for Ad Hoc Networks*

Chao Gui[1] and Baolin Sun[1,2,3]

[1] College of Computer Science & Technology, Hubei University of Economics
Wuhan 430205, P.R. China
blsun@163.com
[2] School of Computer Science and Technology, Wuhan University of Technology
Wuhan 430063, P.R. China
[3] Department of Mathematics and Physics, Wuhan University of Science and Engineering
Wuhan 430073, P.R. China

Abstract. As multimedia and group-oriented computing becomes increasingly popular for the users of mobile ad hoc networks (MANET). Due to the dynamic nature of the network topology and restricted resources, quality of service (QoS) and multicast routing in MANET is a challenging task. It attracts the interests of many people. In this paper, we present a fuzzy controllers based QoS routing algorithm with a multiclass scheme (FQRA) in MANET. The performance of this scheduler is studied using NS2 and evaluated in terms of quantitative met-rics such as path success ratio, average end-to-end delay and throughput. Simu-lation shows that the approach is efficient, promising and applicable in MANET.

1 Introduction

A Mobile Ad hoc NETwork (MANET) is an autonomous system of mobile nodes connected by wireless links. There is no static infrastructure such as base station as that was in cell mobile communication. In ad hoc network, if two nodes are not within radio range, all message communication between them must pass through one or more intermediate nodes. All the nodes are free to move around randomly, thus changing the network topology dynamically [1-5,7,8]. These types of networks have many advantages, such as self-reconfiguration and adaptability to highly variable mobile characteristics like the transmission conditions, propagation channel distribution characteristics and power level. They are useful in many situations such as military applications, conferences, lectures, emergency search, rescue operations and law enforcement. However, such benefits come with new challenges which mainly resides in the unpredictability of the network topology due to mobility of nodes and the limited available bandwidth due to the wireless channel. These characteristics demand a new way of designing and operating this type of networks. For such networks,

[*] This work is supported by National Natural Science Foundation of China (No. 90304018), NSF of Hubei Province of China (No. 2005ABA231, 2006ABA301) and Key Scientific Research Project of Hubei Province Education Department (No. D200617001).

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 506 – 514, 2006.

an effective routing protocol is critical for adapting to node mobility as well as possible channel error to provide a feasible path for data transmission [9-15].

Multicasting is a promising technique to provide a subset of the network with the service it demands while not jeopardizing the bandwidth requirements of others. The advantage of multicasting is that packets are multiplexed only when it is necessary to reach two or more receivers on disjoint paths [1,2,6-7,15]. As a result of their broadcasting capability, ad hoc networks are inherently ready for multicasting. In addition multicast gives robust communication whereby the receiver address is unknown or modifiable without the knowledge of the source within the wireless environment.

Quality of service (QoS) support for multimedia applications is closely related to resource allocation, the objective of which is to decide how to reserve resources such that QoS requirements of all the applications can be satisfied [1-3,6-9]. However, it is a significant technical challenge to provide reliable high-speed end-to-end communications in these networks, due to their dynamic topology, distributed management, and multi-hop connections. The provision of QoS requirements is of utmost importance for the development of future networks. For supporting QoS aware applications, QoS based routing algorithms such as Core extraction dynamic source routing (CEDAR) [7] and Ticket base routing (TBR) [8] are proposed in the literature. Lorenz and Orda demonstrate in [9] that this uncertainty places additional constraints on QoS provisioning. These algorithms determine a path that satisfies the required QoS. The success of these algorithms purely depends on the existence and reliability of that path.

Fuzzy Logic based decision algorithm influences caching decisions of multiple paths uncovered during route discovery and avoids low quality paths [10]. Differentiated resource allocation considering message type and network queue status is evaluated using fuzzy logic scheme [11]. In [4-6], they proposed the use of fuzzy logic controllers for the dynamic reconfiguration of edge and core routers. This reconfiguration allows for adjusting the network provisioning according to the incoming traffic and the QoS level achieved. A fuzzy controller is specified by fuzzy sets definition (membership function) and a set of rules (rule base).

In this paper, we present a Fuzzy controllers based QoS Routing Algorithm with a multiclass scheme (FQRA) in mobile ad hoc networks. The performance of this scheduler is studied using NS2 and evaluated in terms of quantitative metrics such as improved path success ratio, reduced average end-to-end delay and increased throughput.

The rest of the paper is organized as follows. Section 2 introduces the ad hoc net-work model and route issues. Section 3 presents the fuzzy QoS controller. Some simulation results are provided in Section 4. Finally, Section 5 presents the conclu-sions.

2 Network Model and Routing Issues

A network is usually represented as a weighted digraph G = (N, E), where N denotes the set of nodes and E denotes the set of communication links connecting the nodes. |N| and |E| denote the number of nodes and links in the network respectively, Without loss of generality, only digraphs are considered in which there exists at most one link between a pair of ordered nodes.

In $G(N, E)$, considering a QoS constrained multicast routing problem from a source node to multi-destination nodes, namely given a non-empty set $M=\{s, u_1, u_2, \ldots, u_m\}$, MN, s is source node, $U=\{u^{\subseteq}_1, u_2, \ldots, u_m\}$ be a set of destination nodes. In multicast tree $T= (N_T, E_T)$, where $N_T \subseteq N$, $E_T \subseteq E$. if $C(T)$ is the cost of T, $P_T(s,u)$ is the path from source node s to destination node $u \in U$ in T, $D_T(s, u)$ and $B_T(s, u)$ are the delay and usable bandwidth of $P_T(s, u)$.

Definition 1: The cost of multicast tree *T* is:

$$C(T_e) = \sum\nolimits^{e\in E_T} C(e), \; e \in E_T.$$

Definition 2: The bandwidth of multicast tree *T* is the minimum value of link bandwidth in the path from source node s to each destination node $u \in U$. i.e.

$$B_T(s, u) = min\,(B(e), e \in E_T).$$

Definition 3: The delay of multicast tree *T* is the maximum value of delay in the path from source node s to each destination node $u \in U$. i.e.

$$D_T(s, u) = max(\sum\nolimits_{e\in P_T(n_0,u)} D(e),\; u \in U)\cdot \quad .$$

Definition 4: Assume the minimum bandwidth constraint of multicast tree is *B*, the maximum delay constraint is *D*, given a multicast demand R, then, the problem of bandwidth, delay constrained multicast routing is to find a multicast tree *T*, satisfying:

(1) Bandwidth constraint: $B_T(s, u) \geq B$, $u \in U$.

(2) Delay constraint: $D_T(s, u) \leq D$, $u \in U$.

Suppose *S(R)* is the set, *S(R)* satisfies the conditions above, then, the multicast tree *T* which we find is:

$$C(T) = min\,(C\,(T_s), T_s \in S(R))$$

3 Fuzzy QoS Controller

3.1 Fuzzy Logic Controller

The fuzzy logic was introduced by Zadeh [13] as a generalization of the boolean logic. The difference between these logics is that fuzzy set theory provides a form to repre-sent uncertainties, that is, it accepts conditions partially true or partially false. Fuzzy logic is the best logic to treat random uncertainty, i.e., when the prediction of a se-quence of events is not possible.

Fuzzy logic control system is rule-based system in which a set of so-called fuzzy rules represents a control decision mechanism to adjust the effects of certain causes coming from the system. The aim of fuzzy control system is normally to substitute for or replace a skilled human operator with a fuzzy rule-based system. Specifically, based on the current state of a system, an inference engine equipped with a fuzzy rule base determines an on-line decision to adjust the system behavior in order to guarantee that it is optimal in some certain senses.

There are generally two kinds of fuzzy logic controllers. One is feedback controller, which is not suitable for the high performance communication networks. Another one, which is used in this paper, is shown in Fig. 1. The output of the fuzzy logic controller in Fig. 1 is used to tune the controlled system's parameters based on the state of the system. This control mechanism is different from the conventional feedback control and considered as an adaptive control.

The specific features of the fuzzy controller depend on the model under control and performance measurement. However, in principle, in the fuzzy controller we explore the implicit and explicit relationships within the system and subsequently develop the optimal fuzzy control rules as well as a knowledge base.

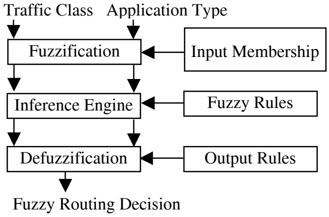


Fig. 1. The fuzzy routing in MANET

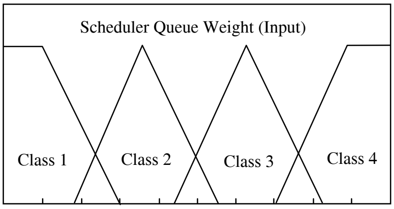


Fig. 2. Scheduler membership functions

3.2 Scheduler Controller

The packet scheduler used in our architecture is WRR (Weighted Round Robin). In this scheduler, queues are served according to a configurable weight that can be changed during network operation. This allows having control of the bandwidth assigned to each service class. The packet delay and discard rate for each queue (class) can be controlled by changing this weight. An example of membership function of schedule controller is showed in Fig. 2. Other membership functions are: packet delay in the expedited forwarding queue and discard rate due to queue overflow in the best-effort class. The output membership functions are also defined as trapezoid functions by the same previous reasons. We used the center of gravity defuzzification method, since it gives better results. The output membership function gives the weights assigned to each class in the WRR scheduler.

3.3 Fuzzy Rule Base

Fuzzy systems reason with multi-valued fuzzy sets instead of crisp sets. The Fuzzy Logic Controller (FLC) (Fig. 1) has two inputs: Residual Bandwidth and Traffic Class and one output: Fuzzy Routing Decision [11, 14].

Mamdani fuzzy-rule based systems constitute of a linguistic description in both the antecedent parts and the consequent parts. Each rule (Table 1) is a description of a condition-action statement that may be clearly interpreted by the users. Rule base is an IF-THEN rule group with fuzzy sets that represents the desired behavior of a fuzzy system. It can be defined in agreement with the administrative policy.

Ri: **IF** x_1 is A_{i1} and … and x_n is A_{in} **THEN** y is C_i, $i = 1, 2, \ldots, L$

where L is the number of fuzzy rules, $x_j \in U_j$, $j = 1, 2, \ldots, n$, are the input variables, y is the output variable, and A_{ij} and C_i are linguistic variables or fuzzy sets for x_j and y respectively. A_{ij} and C_i are characterized by both membership functions.

Inputs are of the form: x_1 *is* A_1*'*, x_2 *is* A_2*'*, …, x_n is A_n*'* where A_1*'*, A_2*'*, …, A_n*'* are fuzzy subsets of U_1, U_2, … U_n, which are the universe of discourse of inputs.

Table 1. Fuzzy rule base – QoS classes and application type

QoS Class	Bandwidth Requirement	Application Type
1	256 Kbps	Non-real-time flow with normal service
2	512 Kbps	Non-real-time flow with preference service
3	2 Mbps	Real-time flow with normal service
4	4 Mbps	Real-time flow with preference service

4 Simulation

4.1 Random Graph Generation

In generating random graphs, we have adopted the method used in [16], where vertices are placed randomly in a rectangular coordinate grid by generating uniformly distributed values for their x and y coordinates. The remaining edges of the graph are chosen by examining each possible edge *(u,v)* and generating a random number $0 \leq r < 1$. If r is less than a probability function *P(u,v)* based on the edge distance between u and v, then the edge is included in the graph. The distance for each edge is the Euclidean distance (denoted as *d(u,v)* between the nodes that form the end-points of the edge. We used the probability

$$P\,(u,v) = \beta\, exp(-\ d(u,v)aL\ )$$

where *d(u,v)* is geometric distance from node *u* to node *v, L* is maximum distance between two nodes. The parameters α and β are in the range (0, 1) and can be used to obtain certain desirable characteristics in the topology, parameter α can be used to control short edge and long edge of the random graph, and parameter β can be used to control the value of average degree of the random graph.

4.2 Simulation Model

To conduct the simulation studies, we have used randomly generated networks on which the algorithms were executed. This ensures that the simulation results are independent of the characteristics of any particular network topology. Using randomly generated network topologies also provides the necessary flexibility to tune various network parameters such as average degree, number of nodes, and number of edges, and to study the effect of these parameters on the performance of the algorithms. The platform used was the Network Simulator (NS), version 2.26 [17].

Our simulation modeled a network of mobile nodes placed randomly within 1000 x 1000 meter area. Each node has a radio propagation range of 250 meters and chan-nel capacity of 5 Mbps. Two-ray propagation model was used. The IEEE 802.11 distributed coordination function was used as the medium access control protocol. A random waypoint mobility model was used: each node randomly selects a position, and moves toward that location with a speed ranging from just above 0 m/s to 10 m/s. When the node reaches that position, it becomes stationary for a programmable pause time; then it selects another position and repeats the process. The simulation was repeated with different seed values. A traffic generator was developed to simulate CBR sources. The size of the data payload is 512 bytes. Data sessions with randomly selected sources and destinations were simulated. Each source transmits data packets at a minimum rate of 4 packets/sec. and maximum rate of 10 packets/sec. Traffic Classes were randomly assigned and simulation was carried out with different bandwidth requirements. There were no network partitions throughout the simulation. Each simulation is executed for 600 seconds of simulation time. Multiple runs with different seed values were conducted for each scenario and collected data was averaged over those runs. Table 2 lists the simulation parameters which are used as default values unless otherwise specified.

Table 2. Simulation parameters

Number of nodes	100
Terrain range	1000m ×1000 m
Transmission range	250 m
Simulation duration	1 h
Speed	0-10 m/s
Mobility model	Random way point
Propagation model	Free space
Channel bandwidth	5 Mbps
Traffic type	CBR
Data payload	512 bytes/packet
Service class distribution	4:2:3:1
Node pause time	0-10 seconds

4.3 Performance Metrics

The following metrics are used in computing the scheduler performance. The metrics were derived from one suggested by the MANET working group for routing protocol evaluation.

- Throughput

The rate of data being received at the servers.

- Average end-to-end delay

This indicates the end-to-end delay experienced by packets from source to destina-tion.

- Path success ratio

Route success ratio is the ratio of the number of total number of connection request discover to the destinations to the number of routed connection requests. This number presents the effectiveness of the protocol.

4.4 Simulation Results

In this performance evaluation the following performance metrics were evaluated: percentile of path success ratio, edge-to-edge delay and throughput. For each evaluation, we used CBR. All simulations start with initial scheduler configuration with 60% of the bandwidth for each class. To eliminate simulation results with an empty network, we start collecting results 30 seconds after the beginning of the simulation.

After optimization procedure was executed, we could verify the result comparing packet delivery ratio, average end-to-end delay and throughput.

Fig. 3 shows the effect of network size on throughput. We can see that non-QoS's throughput is smaller than of FQRA with the increasing of the scale of the network.

Fig. 4 shows the effect of number of nodes over average end-to-end delay. Delay is more in FQRA and can be improved by introducing multiple paths during fuzzy routing and by giving more precedence to the packets which are waiting for their service.

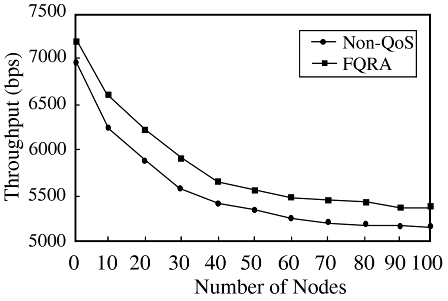


Fig. 3. Throughput vs. Networks size

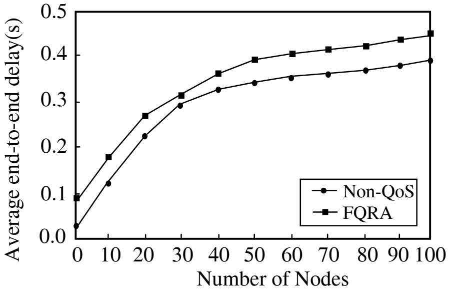


Fig. 4. Average end-to-end delay vs. Networks size

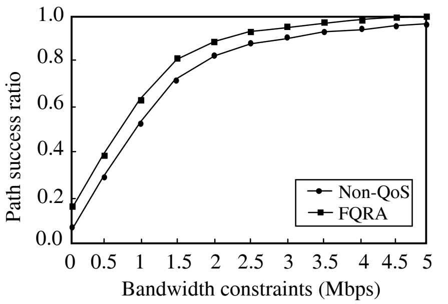


Fig. 5. Path success ratio vs. Bandwidth constraints

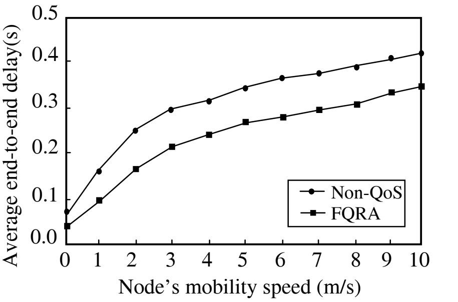


Fig. 6. Average end-to-end delay vs. Node's mobility speed

Fig. 5 depicts a comparison path success rate to find the path through non-QoS and FQRA. With the relaxation of bandwidth constraints, the path success rate becomes higher for non-QoS. The success rate is still higher than that of non-QoS, which means is more suitable for the routing choosing under timely data transmission application and dynamic network structure.

The average and-to-end delay performance as shown in the Fig. 6, proves that the end-to-end delay improves when scheduler is included. As the mobility varies from 0-10 m/s, the fuzzy controllers scheduler provides an end-to-end delay reduced by around 0.01 sec. to 0.05 sec.

5 Conclusion and Future Work

Our QoS routing algorithm has produced significant improvements in throughput, average end-to-end delay and path success ratio. Fuzzy logic implementation relates input and output in linguistic terms, the overlap composition of many input variables (multiple QoS criterion) in taking a single output decision shows the robustness of the system in adapting to constantly changing mobile scenario. The membership functions and rule bases of the fuzzy scheduler are carefully designed. The use of fuzzy logic improves the handling of inaccuracy and uncertainties of the ingress traffic into the domain.

In this paper, we present a fuzzy controllers based QoS routing algorithm with a multiclass scheme in mobile ad hoc networks. The performance of this scheduler is studied using NS2 and evaluated in terms of quantitative metrics such as path success ratio, average end-to-end delay and throughput. Simulation shows that the approach is efficient, promising and applicable in ad hoc networks.

Future work includes comparison with "crisp" versions of the fuzzy algorithm to isolate the contributions of fuzzy logic, as well as applications of fuzzy control to power consumption and directional antennas in MANETs. We also intend to compare FQRA with other QoS routing.

References

1. Sun, B.L., Li, L.Y.: A QoS Multicast Routing Optimization Algorithms Based on Genetic Algorithm. Journal of Communications and Networks, Vol. 8, No. 1, (2006) 116~122
2. Sun, B. L., Li, L. Y., Yang, Q., and Xiang, Y.: An Entropy-Based Stability QoS Multicast Routing Protocol in Ad Hoc Network. Advances in Grid and Pervasive Computing (GPC 2006). Lecture Notes in Computer Science, Vol. 3947, Springer-Verlag Berlin Heidelberg, (2006) 217-226
3. Sun, B. L., Yang, Q., Ma J. and Chen H.: Fuzzy QoS Controllers in Diff-Serv Scheduler using Genetic Algorithms. Computational Intelligence and Security (CIS2005). Lecture Notes in Artificial Intelligence, Vol. 3801, Springer-Verlag Berlin Heidelberg, (2005) 101-106
4. Zhang, R., and Ma, J.: On the Enhancement of a Differentiated Services Scheme. Proceedings IEEE Network Operations and Management Symposium 2000 (NOMS'2000), Honolulu, USA, April (2000) 975-976
5. Zhang, R., Phillis, Y.: Fuzzy Control of Queueing System with Heterogeneius Servers. IEEE Trans. Fuzzy Systems, Vol. 7, No. 1, (1999) 17-26

6. Fernandez, M. P., de Castro, A., Pedroza, P., and de Rezende, J. F.: QoS provisioning across a diffserv domain using policy-based management. in Globecom 2001, San Antonio, USA, Nov. (2001) 2220-2224
7. Sivakumar, R., Sinha, P. and Bharghavan, V.: CEDAR: Core Extraction Distributed Ad hoc Routing. IEEE Journal on Selected Areas in Communication, Vol. 17, No. 8, (1999) 1454-1465
8. S. Chen, S., and Nahrstedt, K.: Distributed Quality of Service routing in Ad hoc networks. IEEE journal on selected Areas in Commn., Vol. 17, No. 8, (1999) 1488-1504
9. Lorenz, D. H., Orda, A.: Qos routing in networks with uncertain parameters. IEEE/ACM Transactions on Networking, Vol. 6, No. 6, (1998) 768-778
10. Rea, S., and Pesch, D.: Multi-metric routing decisions for ad hoc networks using fuzzy logic. In Proceedings of 1st International Symposium on Wireless Communication Systems, Mauritius, 20- 22 September, (2004) 403-407
11. Alandjani, G., and Johnson, E.: Fuzzy Routing in ad hoc networks. In Proceedings of the IEEE International Conference on Performance, Computing and Communications, Phoenix, Arizona, April, (2003) 525-530
12. Gomathy, C., Shanmugavel,S.: An Efficient Fuzzy Based Priority Scheduler for Mobile Ad hoc Networks and Performance Analysis for Various Mobility Models. 2004 IEEE Wireless Communications and Networking Conference (WCNC 2004), Atlanta, USA, Vol. 2, March (2004) 1087-1092
13. Zadeh, L. A.: Fuzzy sets. Information and Control, vol. 8, (1965) 338-353
14. Thomas, A., Chellappan, C., and Jayakumar, C.: ANTHOC - QoS: Quality of Service Routing in Mobile Ad Hoc Networks using Swarm Intelligence. The Second Asia Pacific Conference on Mobile Technology, Applications and Systems, Guangzhou, China, 15- 17 November, (2005) 8 Pages
15. Sun, Q., Li, L. Y.: An Efficient Distributed Broadcasting Algorithm for Ad Hoc Networks. Advanced Parallel Processing Technologies (APPT 2005), Lecture Notes in Computer Science, Vol. 3756, Springer Verlag Berlin Heidelberg, (2005) 363-372
16. Waxman, B.: Routing of Multipoint Connections. IEEE Journal on Selected Areas in Communications, No. 6, (1988) 1617-1622
17. The Network Simulator - ns-2,: http://www.isi.edu/nsnam/ns/. (2004)

Direction of Arrival Estimation Based on Minor Component Analysis Approach

Donghai Li, Shihai Gao, Feng Wang, and Fankun Meng

Zhengzhou Information Science and Technology Institute
Zhengzhou, China
ldhai99@yahoo.com.cn

Abstract. Many high resolution DOA estimation algorithms like MUSIC and ESPRIT estimation are based on the sub-space concept and require the eigen-decomposition of the input correlation matrix. As quantities of computation of eigen-decomposition, it is unsuitable for real time processing. An algorithm for noise subspace estimation based on minor component analysis is proposed. These algorithms are based on anti-Hebbian learning neural network and contain only relatively simple operations, which are stable, convergent, and have self-organizing properties. Finally a method of real-time parallel processing is proposed, and data processing can be finished at end time of sampling. Simulations show that the proposed algorithm has an analogy performance with the MUSIC algorithm.

1 Introduction

Most of the antenna array direction of arrival(DOA) estimation methods are based on the sub-space concept and require the eigen-decomposition of the input correlation matrix. State-space method [2], MUSIC [3], ESPRIT[4], and Min-Norm [5] are examples of these techniques. Based on the eigen-decomposition of covariance matrix of the array output, they offer high resolution and give accurate estimates. A key limitation of these techniques is the computational burden to process a new sample (snapshot), so they are unsuitable for real time applications. Some attempts have been made to reduce the computational burden of these methods[1][6][7][8]. In this work, An algorithm for noise subspace estimation based on minor component analysis is proposed. These algorithms are based on anti-Hebbian learning neural network and contain only relatively simple operations, These algorithms are stable, convergent, and have self-organizing properties[9][10]. Finally a method of Real-time parallel processing is proposed, at end time of sampling data, processing also can be finished. Simulations show that the proposed algorithm has an analogy performance with MUSIC algorithm.

2 DOA Principle

Most of the antenna array DOA estimation methods are based on the sub-space concept and require the eigen-decomposition of the input correlation matrix.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 515 – 522, 2006.

2.1 DOA Data Model

If there are p signals incident onto the array, the received input data vector at an M-element array can be expressed as a linear combination of the p incident waveforms and noises.

If $x(t)$ is the array element received signal, $s(t)$ is source signal, $w(t)$ is additive noise, the first array element is taken as the reference array element, then the *kth* array element received signal is

$$x_k(t) = \sum_{i=1}^{p} s_i(t)a(\theta_i) + w_k(t) \tag{1}$$

where, $k = 1,2,\cdots M$, the vector form is as follow:

$$x(t) = \mathbf{As}(t) + \mathbf{w}(t) \tag{2}$$

where : $x(t) = [x_1(t),\ x_2(t),\ldots\ldots\ x_M\ (t)]^T_{M\times 1}$

A is the matrix of steering vectors

$$\mathbf{A} = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \ldots\ldots \mathbf{a}(\theta_P)]_{M\times P} \tag{3}$$

$\mathbf{s}(t) = [s_1(t), s_2(t), \ldots\ldots s_p(t)]^T_{p\times 1}$ is the signal vector, and $\mathrm{w}(t) = [w_1(t), w_2(t), \ldots\ldots w_M(t)]^T_{M\times 1}$ is a noise vector with components of variance σ_n^2.

The received vectors and the steering vectors can be visualized as vectors in an M-dimensional vector space.

2.2 MUSIC Algorithm

The input covariance matrix is

$$R_{xx} = E[XX^H] = AR_{SS}A^H + \sigma_n^2 I \tag{4}$$

where R_{ss} is the signal correlation matrix. I is an identity matrix of appropriate dimension, and $(.)^H$ denotes conjugate transpose.

The eigen-decomposition of the positive definite Hermitian matrix R_{xx} is given by

$$R_{xx} = \sum_{i=1}^{P} \lambda_l e_i e_i^H + \sigma_n^2 \sum_{i=p+1}^{M} e_i e_i^H \tag{5}$$

where λ_l is the eigenvalue corresponding to the eigenvector e_i, stored in decreasing order, for all $i=1 \ldots M$.

The eigenvectors of the covariance matrix R_{xx} belong to either of the two orthogonal subspaces, the principal eigen subspace (signal subspace) and the minor eigen subspace (noise subspace).

The dimension of the signal subspace is p, while the dimension of the noise subspace is M-p.

The M-p smallest eigenvalues of R_{xx} are equal to σ_n^2, and the eigenvectors e_i, *i=p+1, ... ,M*, corresponding to these eigenvalues span the noise subspace.

The p steering vectors that make up A lie in the signal subspace and are hence orthogonal to the noise subspace.

By searching through all possible array steering vectors to find those which are orthogonal to the space spanned by the noise eigenvectors e_i, *i=p+1, ... ,M*, the DOAs $\theta_1, \theta_2, ...\theta_P$, can be determined.

To form the noise subspace, we form a matrix $\mathbf{V}_n$ containing the noise eigenvectors e_i, *i=p+1, ... ,M.*

Then $\mathbf{a}^H(\theta)\mathbf{V}_n\mathbf{V}_n^H\mathbf{a}(\theta) = 0$ for θ corresponding to the DOA of a multiple component.

The DOAs of the multiple incident signals can be estimated by locating the peaks of a MUSIC spatial spectrum

$$P(\theta) = \frac{1}{\mathbf{a}^H(\theta)\mathbf{V}_n\mathbf{V}_n^H\mathbf{a}(\theta)} \tag{6}$$

The resolution of MUSIC is very high even in low SNR.

3 Extracting Multiple Minor Components

Extracting multiple minor components is based on anti-Hebbian learning neural network, it contains only relatively simple operations, which is stable, convergent, and have self-organizing properties.

3.1 Hebbian Learning

A self-organizing principle was proposed by Hebb in 1949 in the context of biological neurons Hebb's principle. When a neuron repeatedly excites another neuron, then the threshold of the latter neuron is decreased, or the synaptic weight between the neurons is increased, in effect increasing the likelihood of the second neuron to excite.

3.2 Generalized Hebbian Algorithm (GHA) [9]

The step of extracting principal components analysis by GHA is as follows:

1. Subtract the contribution of the first principal component.
2. Drive the difference into another Hebbian neuron.
3. This extracts the next principal component.
4. Subtract its contribution. Goto step 2.

With N Hebbian neurons, we'll get all N principal components.
Embodied in Sanger's rule:˘

$$W_{k+1} = W_k + \Delta W_k \quad \Delta w_{ij} = \eta V_i \left(\xi_j - \sum_{k=1}^{i} V_k w_{kj} \right) \tag{7}$$

Sanger's rule in action based on the Hebbian Neuron, revisited is explained as Fig 1.

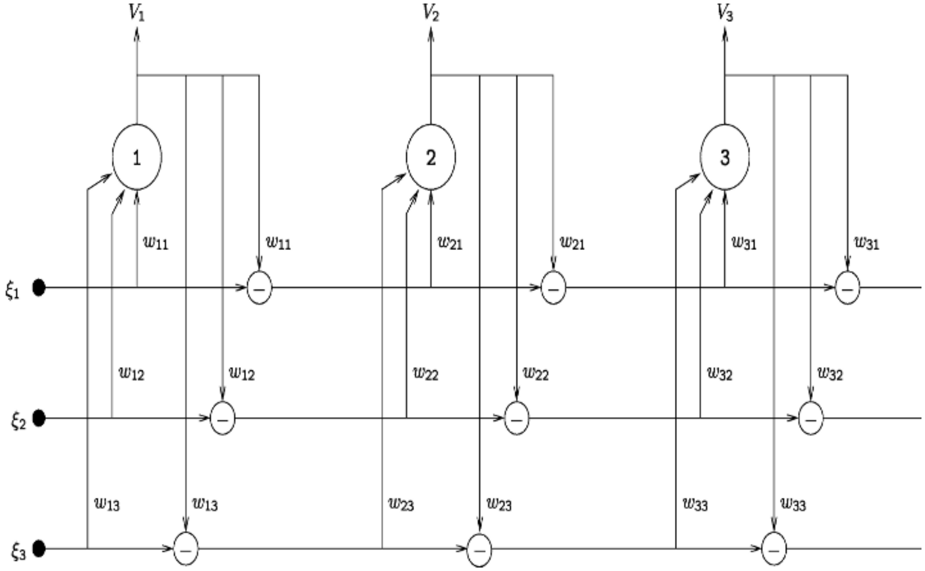


Fig. 1. Cascading multiple Hebbian neurons

3.3 Extracting Multiple Minor Components[10]

Modifying the GHA rule is as follows:

$$W_{k+1} = W_k + \Delta W_k \quad \Delta w_{ij} = -\eta V_i \left(\xi_j - \sum_{k=1}^{i} V_k w_{kj} \right) \tag{8}$$

It is anti-Hebbian network, the anti-Hebbian rule find the direction in space that has the minimum variance. In other words, it can extract multiple minor components.

Anti-Hebbian does de-correlation, which de-correlates the output from the input.

Hebbian rule is unstable, since it tries to maximize the variance. Anti-Hebbian rule, on the other hand, is stable and convergent.

3.4 Complex Signal Processing

$x(t)$ is complex-value data. To convert it into a real-value modal, the complex vectors e_k, k = 1, . . . ,M and R_{xx} should first be decomposed into their real and imaginary constituents as follows [1]:

$$e_k = e_{kr} + je_{ki} \text{ and } R = R_r + jR_i \tag{9}$$

Then the equation becomes:

$$(R_r + jR_i)(e_{kr} + je_{ki}) = \lambda_k (e_{kr} + je_{ki}) \tag{10}$$

It equivalently is :

$$R_r e_{kr} - R_i e_{ki} = \lambda_k e_{kr}, R_i e_{kr} + R_r e_{ki} = \lambda_k e_{ki} \tag{11}$$

Moreover, by combining terms, we get:

$$\begin{bmatrix} R_r & -R_i \\ R_i & R_r \end{bmatrix} \bullet \begin{bmatrix} e_{kr} \\ e_{ki} \end{bmatrix} = \lambda_k \begin{bmatrix} e_{kr} \\ e_{ki} \end{bmatrix}$$

$$R_c w_k = \lambda_k w_k \text{ , } R_c = \begin{bmatrix} R_r & -R_i \\ R_i & R_r \end{bmatrix}, \; w_k = \begin{bmatrix} e_{kr} \\ e_{ki} \end{bmatrix} \tag{12}$$

$$R_r = E[x_r(t)x_r^T(t)] + E[x_i(t)x_i^T(t)] \text{ , } R_i = E[x_i(t)x_r^T(t)] - E[x_r(t)x_i^T(t)]$$

$$X_c = \begin{bmatrix} x_r & -x_i \\ x_i & x_r \end{bmatrix} \tag{13}$$

3.5 Real-Time Parallel Processing

Can be seen from the equation(13), when the sampling data is finished, the calculation has not been finished, there is a need of computation for data $\begin{bmatrix} x_r \\ x_i \end{bmatrix}$.As this neural network algorithms is not relation of data sequence, we change the data sequence, and equation(13) can be changed to equation(14).

$$X_c = \begin{bmatrix} x_{r1} & -x_{i1} & x_{r2} & -x_{i2} & \cdots & x_{rN} & -x_{iN} \\ x_{i1} & x_{r1} & x_{i2} & x_{r2} & \cdots & x_{iN} & x_{rN} \end{bmatrix} \tag{14}$$

Where N is numbers of snapshot.

Using this model, when dynamic sampling data is end, the calculation of this algorithms also can be finished, so it can be used for the real-time processing.

4 Simulations

In this section, we present some simulation results illustrating the properties of the proposed approach. In all examples, we use a uniform linear array with 16 elements spaced λ/2 apart, where λ denotes the wavelength of the sources signals.

There are three signals, angles are 30,50,70 degree, data input sequence is as equation(14),numbers of snapshot is 2048, We have simulated the above iterative procedure using the learning rule (8), the number of iterations is 1 ,with rate η =0.005 and initial weights matrix W=0.5*I; to extract the noise subspace. Fig. **2** gives the result by MUSIC at 10dB, Fig.**3** gives the result by neural network at 10dB, and Fig. **4** gives the results by MUSIC at 30dB, Fig.**5** gives the results by neural network at 30dB.

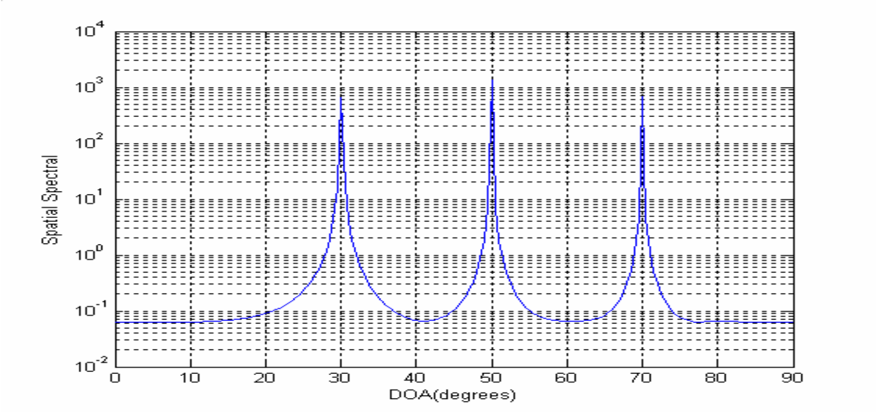


Fig. 2. DOA estimation by MUSIC at 10dB

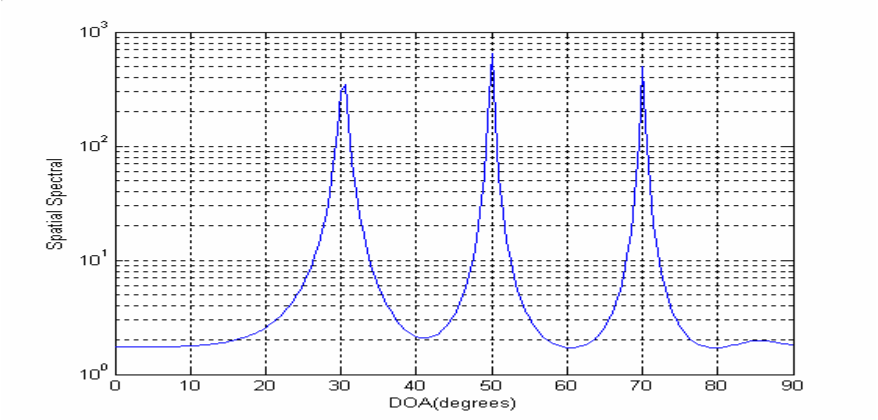


Fig. 3. DOA estimation by neural network at 10dB

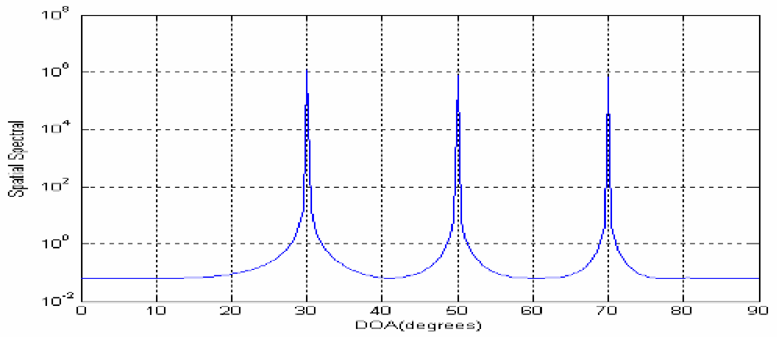


Fig. 4. DOA estimation by MUSIC at 30dB

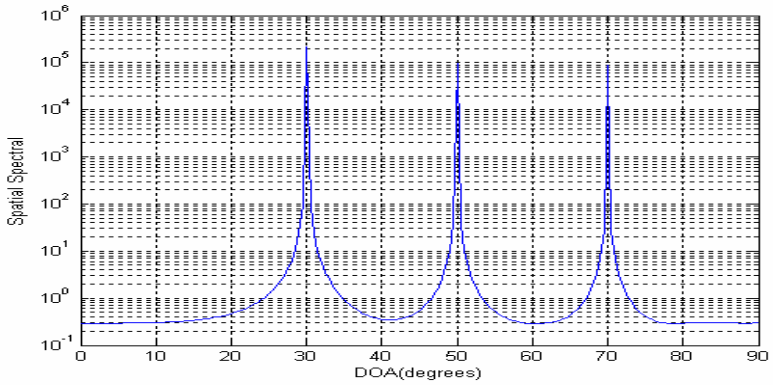


Fig. 5. DOA estimation by neural network at 30dB

5 Conclusion

An algorithm for noise subspace estimation based on minor component analysis is proposed in this paper. These algorithms are based on anti-Hebbian learning neural network and contain only relatively simple operations, These algorithms are stable and convergent. A method of real-time parallel processing is proposed, and data processing can be finished at end time of sampling. Simulated results show that the proposed algorithm be of an analogy performance with MUSIC algorithm.

References

1. L. Badidi, L. Radouane, A neural network approach for DOA estimation and tracking, Proceedings of the 10th IEEE Workshop on Statistical Signal and Array Processing, Pocono Manor, PA, USA, August 2000, pp. 434-438.
2. R. J. Vaccaro and Y. Ding, "A new state-space approach for direction finding" IEEE Trans. Signal Processing, vol. 42, no. 11, pp. 3234-3237, Nov. 1994.

3. R. 0. Schmidt, "Multiple emitter location and signal parameter estimation," IEEE Trans. Antennas Propaga., vol. 34, no. 3, pp. 276-280, Mar. 1986.
4. R. Roy and T. Kailath, “ESPRIT-estimation of signal parameters viarotational invariance techniques,” IEEE Trans. Acoust., Speech, SignalProcessing, vol.37, pp.984–995, July 1987.
5. R. Kumaresan and D. W. Tufts,"Estimating the angles of arrival of multiple source plane waves" ,IEEE Trans. Aerosp. Elect. Syst., vol. AES-19, pp. 134-149,January, 1983.
6. G. W. Stewart, "An updating algorithm for subspace tracking," IEEE Trans. Signal Processing, vol. 40, pp. 1535-1541, 1992.
7. A. Eriksson, P. Stoica, and T. Soderstom, "On-line subspace algorithms for tracking moving sources," IEEE Trans. Signal Processing, vol. 42, no. 9, pp.2319-2330, Sept. 1994.
8. C. C. Yeh, "Simple computation of projection matrix for bearing estimation," Proc. Inst. Elec. Eng., Part F,vol. 134, pp.146-150, Apr. 1987.
9. T.D. Sanger, Optimal Unsupervised Learning in a Single-Layer Neural Network, Neural Networks, Vol. 2, pp. 459 - 473, 1989.
10. F. Palmieri,J.Zhu,and C. Chang, Anti-Hebbian learning in topo-logically constrained linear networks: A tutorial, IEEE Trans.on Neural Networks,Vol. 4,No. 5,pp. 748 -761, Sept. 1993.

Two-Stage Temporally Correlated Source Extraction Algorithm with Its Application in Extraction of Event-Related Potentials

Zhi-Lin Zhang[1,2], Liqing Zhang[1], Xiu-Ling Wu[1], Jie Li[1], and Qibin Zhao[1]

[1] Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China
[2] School of Computer Science and Engineering,
University of Electronic Science and Technology of China,
Chengdu 610054, China
zlzhang@uestc.edu.cn, zhang-lq@cs.sjtu.edu.cn

Abstract. To extract source signals with certain temporal structures, such as periodicity, we propose a two-stage extraction algorithm. Its first stage uses the autocorrelation property of the desired source signal, and the second stage exploits the independence assumption. The algorithm is suitable to extract periodic or quasi-periodic source signals, without requiring that they have distinct periods. It outperforms many existing algorithms in many aspects, confirmed by simulations. Finally, we use the proposed algorithm to extract the components of visual event-related potentials evoked by three geometrical figure stimuli, and the classification accuracy based on the extracted components achieves 93.2%.

1 Introduction

It is known that blind source extraction (BSE) algorithms are suitable for extracting a few of temporally correlated source signals from large numbers of sensor signals, say recordings of 128 EEG sensors [1]. In practice they require certain additional *a priori* information of the desired source signals. Thus they generally are implemented in a semi-blind way [2,3,5,6,7].

Among the extraction algorithms there are two famous algorithms, i.e. the cICA algorithm [5] and the FICAR algorithm [6], both of which need to design a so-called reference signal that is closely related to the desired underlying source signal. That is to say, the phase and the morphology of the reference must be matched to that of the desired signal to great extent, or the occurrence time of each impulse of the reference signal is consistent with that of the desired signal [8]. However, in some applications it is difficult to design such a reference, especially when the morphology and the phase of the desired source signals are not expected [3].

Based on our previous primary work [2,3], in this paper we propose a Temporally Correlated signal Extraction algorithm (TCExt algorithm), which does not need the reference, unlike the cICA algorithm and the FICAR algorithm.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 523–532, 2006.

Computer simulations on artificially generated data and experiments on the extraction of event-related potentials show its many advantages.

2 Problem Statement

Suppose that the unknown source signals $\mathbf{s}(k) = [s_1(k), \cdots, s_n(k)]^T$ are mutually statistically independent with zero mean and unit variance, holding the basic simultaneous mixture ICA model [1]. Without lose of generality, we further assume s_1 is the desired temporally correlated source signal, satisfying the following relationship:

$$\begin{cases} E\{s_1(k)s_1(k-\tau^*)\} > 0 \\ E\{s_j(k)s_j(k-\tau^*)\} = 0 \quad \forall j \neq 1 \end{cases} \tag{1}$$

where s_j are other source signals, and τ^* is the optimal lag defined below:

Definition 1. *The non-zero τ^* is called the optimal lag, if the delayed autocorrelation at τ^* of the desired source signal s_1 is non-zero, while the delayed autocorrelation at τ^* of other source signals is zero. Here all of the source signals are supposed to be mutually independent.*

In addition, we give the definition of the optimal weight vector as follows:

Definition 2. *The column vector $\boldsymbol{w}^*$ is called the optimal weight vector of the desired source signal $\boldsymbol{s}_1$, if the following relationship holds:*

$$(\boldsymbol{w}^*)^T \boldsymbol{VAs} = c\boldsymbol{s}_1, \tag{2}$$

where c is a non-zero constant, $\boldsymbol{V}$ is a whitening matrix, and $\boldsymbol{A}$ is a mixing matrix.

3 Framework of the Proposed Algorithm

Based on the assumptions in the previous section, we have proposed a two-stage extraction algorithm framework [3], shown in Fig.1. The first stage is called the capture stage. In this stage, the algorithm coarsely extracts the desired source signal by using correlation information. At the end of the stage, we obtain the weight vector $\hat{\mathbf{w}}$. But it can be shown that due to some practical issues [2,3] $\hat{\mathbf{w}}$ is only close to the optimal weight vector $\mathbf{w}^*$. Therefore the captured source signal $\hat{y} = \hat{\mathbf{w}}^T\mathbf{x}$ is still mixed by the "cross-talk noise".

Next, in the second stage, we exploit the independence assumption and use the output of the first stage, i.e. $\hat{\mathbf{w}}$. At the end of this stage, we obtain a suboptimal solution $\bar{\mathbf{w}}$ [1], which is much closer to $\mathbf{w}^*$ than $\hat{\mathbf{w}}$ is. Thus we finally obtain the desired source signal $\bar{y} = \bar{\mathbf{w}}^T\mathbf{x}$, which is almost not mixed by the "cross-talk noise".

In the framework we will propose an improved extraction algorithm with higher performance, even if the desired source signals have the same period or are near Gaussian.

[1] Note that in practice we almost cannot obtain the optimal solution $\mathbf{w}^*$.

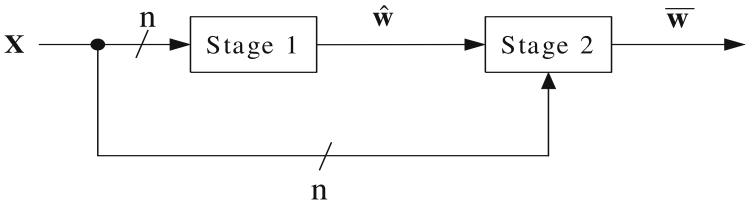


Fig. 1. The framework of the proposed algorithm

3.1 Finding Lags

In practice we cannot find the optimal lag, and we can only find lags that satisfy $E\{s_1(k)s_1(k-\tau_i)\} > E\{s_j(k)s_j(k-\tau_i)\}, \ \forall j \neq 1, \ i = 1, \cdots, P$. Thus, due to performance consideration [2] we suggest to select several suitable lags that correspond to the time structure of the desire source signal, instead of selecting only one lag. For example, for a periodic signal we select the lags corresponding to its fundamental period and multiple periods. The use of several lags, instead of only one lag, can improve the extraction performance, as shown in [3].

There are many methods for finding these lags or the temporal structures [7]. For example, the cepstrally transformed discrete cosine transform [11] can be used to detect the periods of source signals, even if the strengths of signals differ by about 60 dB. In addition, in some applications, such as biomedical signal processing, the lags are often readily available [7,8].

3.2 The First Stage: Coarse Recovery

After choosing suitable lags $\tau_i (i = 1, \cdots, P)$ and whitening the original observations, the first stage employs our previously proposed algorithm [2] to obtain the weight vector $\hat{\mathbf{w}}$:

$$\hat{\mathbf{w}} = EIG\Big(\sum_{i=1}^{P}\Big(\mathbf{R}_{\mathbf{Z}}(\tau_i) + \mathbf{R}_{\mathbf{Z}}(\tau_i)^T\Big)\Big), \tag{3}$$

where $\mathbf{R}_{\mathbf{Z}}(\tau_i) = E\{\mathbf{z}(k)\mathbf{z}(k-\tau_i)^T\}$, $\mathbf{z}(k)$ are the whitened observations, and $EIG(\mathbf{Q})$ is the operator that calculates the normalized eigenvector corresponding to the maximal eigenvalue of the matrix $\mathbf{Q}$.

If the desired signal is periodic, then the algorithm (3) can be rewritten as

$$\hat{\mathbf{w}} = EIG\Big(\sum_{i=1}^{P}\Big(\mathbf{R}_{\mathbf{Z}}(i\tau) + \mathbf{R}_{\mathbf{Z}}(i\tau)^T\Big)\Big), \tag{4}$$

where τ is the fundamental period of the desired source signal. If several desired source signals have the same period, they still can be extracted under some weak conditions, confirmed by the following theorem.

Theorem 1. *Suppose there are q source signals $(s_1, \cdots, s_q)$ that are mutually uncorrelated and have the same period N, and also suppose their autocorrelations satisfy $E\{s_i(k)s_i(k-N)\} \neq E\{s_j(k)s_j(k-N)\}$, $\forall i \neq j$ and $1 \leq i, j \leq q$. Without lose of generality, further suppose $r_1 > \cdots > r_q$, where $r_i = E\{s_i(k)s_i(k-N)\}$. Then the i-th source signal can be perfectly extracted by the weight vector $\boldsymbol{w}_i$ that is the normalized eigenvector corresponding to the i-th largest eigenvalue of $E\{\boldsymbol{z}(k)\boldsymbol{z}(k-N)^T\}$.*

Proof: Since $\mathbf{w}_i$ is the normalized eigenvector corresponding to the i-th largest eigenvalue of $E\{\mathbf{z}(k)\mathbf{z}(k-N)^T\}$, we have $E\{\mathbf{z}(k)\mathbf{z}(k-N)^T\}\mathbf{w}_i = \lambda_i \mathbf{w}_i$, $i = 1, \cdots, q$, where λ_i is the i-th largest eigenvalue. In other words, $\mathbf{VA}E\{\mathbf{s}(k)\mathbf{s}(k-N)^T\}\mathbf{A}^T\mathbf{V}^T\mathbf{w}_i = \lambda_i \mathbf{w}_i$. Since $\mathbf{VA}$ is an orthogonal matrix, then $E\{\mathbf{s}(k)\mathbf{s}(k-N)^T\}(\mathbf{A}^T\mathbf{V}^T\mathbf{w}_i) = \lambda_i(\mathbf{A}^T\mathbf{V}^T\mathbf{w}_i)$, indicating that $(\mathbf{A}^T\mathbf{V}^T\mathbf{w}_i)$ is the normalized eigenvector corresponding to the eigenvalue λ_i of $E\{\mathbf{s}(k)\mathbf{s}(k-N)^T\}$. Due to the distinction of the eigenvalues of $E\{\mathbf{s}(k)\mathbf{s}(k-N)^T\}$, we can deduce that λ_i is its i-th largest eigenvalue, i.e., $\lambda_i = r_i$. According to the assumptions and the previous development, $E\{\mathbf{s}(k)\mathbf{s}(k-N)^T\}$ is a diagonal matrix, and thus we have $(\mathbf{A}^T\mathbf{V}^T\mathbf{w}_i) = \mathbf{e}_i$, whose the i-th element is one while other elements are zero. On the other hand, we have $y = \mathbf{w}_i^T\mathbf{z} = \mathbf{w}_i^T\mathbf{VAs} = \mathbf{e}_i^T\mathbf{s}$, implying the i-th source signal is perfectly extracted. ■

The algorithm (3) has many advantages (see [2,3] for details). However, although it can achieve good extraction quality, it can be shown that the algorithm is insufficient to perfectly recover the desired source signal, and that the solution $\widehat{\mathbf{w}}$ in this stage is just close to the optimal weight vector $\mathbf{w}^*$ [3]. Thus, to make the solution $\widehat{\mathbf{w}}$ further closer to $\mathbf{w}^*$, in the second stage we derive a higher-order statistics based algorithm.

3.3 The Second Stage: Fine Extraction

Under the constraint $\|\mathbf{w}\| = 1$, the maximum likelihood criteria for extracting one source signal is given by

$$\begin{cases} \min & l(\mathbf{w}) = -E\{\log p(\mathbf{w}^T\mathbf{z}(k))\} \\ s.t. & \|\mathbf{w}\| = 1 \end{cases} \tag{5}$$

where $p(\cdot)$ denotes the probability density function (pdf) of the desired source signal. Note that minimizing (5) only leads to one source signal, but not necessarily the desired source signal s_1. However, if we use the $\widehat{\mathbf{w}}$ from the first stage as the initial value, then we can necessarily obtain the s_1.

By the Newton optimization method, we obtain the following algorithm for extracting the desired source signal s_1:

$$\begin{cases} \mathbf{w}^+ = \mathbf{w} - \mu E\{f(\mathbf{w}^T\mathbf{z})\mathbf{z}\}/E\{f'(\mathbf{w}^T\mathbf{z})\} \\ \mathbf{w} = \mathbf{w}^+/\|\mathbf{w}^+\|, \end{cases} \tag{6}$$

with the initial value $\mathbf{w}(0) = \widehat{\mathbf{w}}$. μ is a step-size that may change with the iteration count. In particular, it is often a good strategy to start with $\mu = 1$. $f(\cdot)$ is a nonlinearity, given by $f(\cdot) = -(\log p(\cdot))' = -p(\cdot)'/p(\cdot)$.

In general, the pdf p is unknown and should be estimated. We present a density model that combines the t-distribution density model, the generalized Gaussian distribution density model and the Pearson system model. Our motivation is that the nonlinearity derived from the t-distribution is more robust to the outliers and avoids the stability problem [9], and that the nonlinearity derived from Pearson system can achieve good performance when the desired source signals are skewed and/or near Gaussian [10].

We use the t-distribution [9] to model the super-Gaussian distribution. The derived nonlinearity is

$$f(y) = -p(y)'/p(y) = \frac{(1+\beta)y}{y^2 + \frac{\beta}{\lambda^2}}. \tag{7}$$

where parameters β and λ^2 can be calculated by $\lambda^2 = \beta\Gamma(\frac{\beta-2}{2})/(2m_2\Gamma(\frac{\beta}{2}))$ and $\kappa_t = \frac{m_4}{m_2^2} - 3 = 3\Gamma(\frac{\beta-4}{2})\Gamma(\frac{\beta}{2})/(\Gamma(\frac{\beta-2}{2})^2) - 3$, where m_2 and m_4 are respectively the second-order moment and the fourth-order moment of the distribution. It is clear to see that the function $f(y)$ approaches to zero when the value of y abruptly increases, implying that it is robust to the undue influence of outliers.

To extract the sub-Gaussian source signal, we use the well-known fixed nonlinearity

$$f(y) = y^3, \tag{8}$$

which belongs to the generalized Gaussian density model.

In some applications the desired source signals are skewed, such as the components of the ECG with absolute skewness ranging from 1 to 10. In addition, in some cases the desired source signals are close to Gaussian. Due to these facts, we use the Pearson system to derive a family nonlinearities that are more suitable to extract the skewed and/or near Gaussian signals than the ones derived from the t-distribution and the generalized Gaussian distribution.

The nonlinearity derived from the Pearson system is given by [10]

$$f(y) = -\frac{p_p'(y)}{p_p(y)} = -\frac{(y-a)}{b_0 + b_1 y + b_2 y^2}, \tag{9}$$

where a, b_0, b_1 and b_2 are the parameters of the distribution, calculated by $a = b_1 = -m_3(m_4 + 3m_2^2)/C, b_0 = -m_2(4m_2m_4 - 3m_3^2)/C, b_2 = -(2m_2m_4 - 3m_3^2 - 6m_2^3/C$, where $C = 10m_4m_2 - 12m_3^2 - 18m_2^3$. Note that this type of nonlinearity is also robust to the outliers, just as the nonlinearity given in (7).

Now we have presented three types of nonlinearities for three types of signals. According to the estimated moments, the algorithm (6) adopts suitable nonlinearities. A procedure for the adaptive nonlinearity selection using the sample moments may be given as follows.

Repeat until convergence:

1. Calculate the second, third and fourth sample moments $\hat{m}_2, \hat{m}_3, \hat{m}_4$ for current data $\mathbf{y}(l) = \mathbf{w}^T(l)\mathbf{z}$, where l represents the iteration number.

2. According to the estimated moments, select the nonlinearity as follows:
   - If $\hat{m}_4 > \hat{m}_3^2 + 4.5$, then calculate the nonlinearity (7);
   - if $\hat{m}_4 < 2.5$, then use the nonlinearity (8);
   - if $2.5 \leq \hat{m}_4 \leq \hat{m}_3^2 + 4.5$, then calculate the nonlinearity (9).
3. Calculate the weight vector $\mathbf{w}(l+1)$ using the algorithm (6).

4 Simulations

In the first simulation, we generated seven zero-mean and unit-variance source signals, shown in Fig.2. Each signal had 2000 samples, and its statistics property is shown in Table 1. These signals were randomly mixed and whitened. Our goal was to extract the temporally correlated source signals s_1, s_2, s_3, s_6 and s_7 one by one. After estimated the suitable lags for extracting each desired signal, we employed our proposed two-stage algorithm (TCExt). To make comparisons, we also employed the akExt algorithm [4], the cICA algorithm [5], the FICAR algorithm [6], the SOS algorithm [7], the CPursuit algorithm [13], the SOBI algorithm [12] and the pBSS algorithm [14] on the whitened signals. Note that, in this simulation both the cICA and the FICAR could not extract the source signals due to the difficulty to design the reference signals, but in order to compare the extraction quality, we designed suitable reference signals in advance according to the waveforms of the source signals. To compare the extraction performance we used the following performance index

$$PI = -10E\{lg(s(k) - \tilde{s}(k))^2\} \ (dB) \tag{10}$$

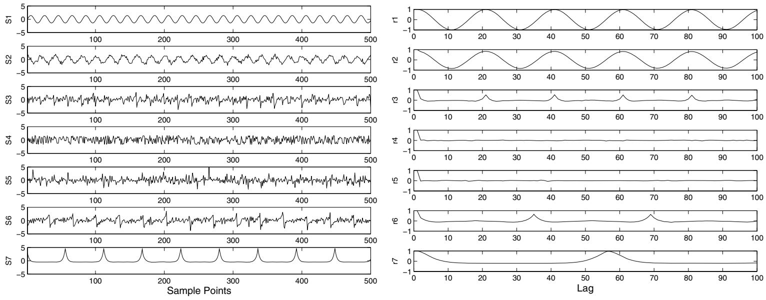
where $s(k)$ is the desired source signal, and $\tilde{s}(k)$ is the extracted signal (both of them are normalized to be zero-mean and unit-variance). The higher PI is, the better the performance is. The averaged performance indexes over 100 independent trials of each algorithm are shown in Table 2, from which we can see that the proposed algorithm generally has better performance than the other algorithms.

Table 1. The properties of the source signals in Fig.2. 'p' denotes the corresponding signal was strictly periodic; 'c' denotes temporally correlated but not strictly periodic; 'n' denotes random noise without any time structure.

source signal	s_1	s_2	s_3	s_4	s_5	s_6	s_7
periodicity	p	c	c	n	n	c	p
kurtosis	-1.5	-1.0	0.7	-1.2	2.8	0.4	7.5

In the next experiment we applied our algorithm to extract potentials evoked by three types of geometrical figures stimuli, and our objective is to classify each type of figures according to the extracted visual evoked potentials (VEPs).

One right-handed subject, aged 21, volunteered to participate in the present study. The subject was healthy both in psychological and neurological, and had



(a) (b)

Fig. 2. Source signals. (a) A segment of the seven source signals. Note that s_1, s_2, s_3 had the same period, but differ in autocorrelations. (b) The corresponding autocorrelation functions of the source signals of (a).

Table 2. The averaged performance index of each algorithm over 100 independent trials. '-' indicates that PI was less than 5 dB or the algorithm could not converge in all the trials. 'akExt(1)' indicates that the value of the parameter τ of the akExt algorithm was equal to the fundamental period of the desired signal; 'akExt(2)' indicates the value of τ was equal to the doubled fundamental period. The same with 'SOS(1)' and 'SOS(2)'.

	TCExt	akExt(1)	akExt(2)	SOS(1)	SOS(2)	cICA	FICAR	SOBI	CPursuit	pBSS
PI of s_1	48.0	17.6	15.9	41.2	37.3	20.6	13.2	8.9	48.3	7.9
PI of s_2	26.9	-	-	-	-	-	-	8.7	27.8	10.2
PI of s_3	12.2	-	-	-	-	-	-	14.6	10.7	6.7
PI of s_6	22.0	34.9	-	-	-	-	20.9	32.2	21.4	-
PI of s_7	57.3	42.0	37.7	45.6	41.4	39.4	34.3	35.7	36.2	20.1

a normal vision. He was seated in a comfort and fixed chair, 0.7m far from the screen of monitor, in a sound and light attenuated RF shielded room.

Three types of geometrical figures(five different-size units for each type) were presented to the subject, i.e. the circle, the square, and the triangle figures. In each trial, a type of geometrical figures, say the circle figure, appeared according to the sequence illustrated in Fig.3. In order to reduce the subject's expectation, each trial showed a type of figures randomly (each type was showed in identical probability). EEG signals were recorded (see Fig.4), sampled at 1000 Hz (thus each trial had 3174 samples) and bandpass filtered between 0.1 Hz and 200 Hz, by a 64-channel EEG system (SynAmps2, Neuroscan, at our Lab for Perception Computing at Shanghai Jiao Tong University, China).

From the original EEG data, we used the proposed algorithm to extract three VEP components by the following procedure. Suppose we had extracted q ($q < 3$) components of VEPs, which corresponded to the first q largest eigenvalues among

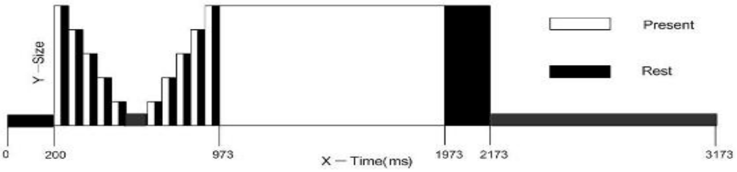


Fig. 3. A stimuli sequence in one trial. The X axis showed the lasting time of the presence or the non-presence of the figure stimuli. The Y axis showed the relative size of the geometrical figure.

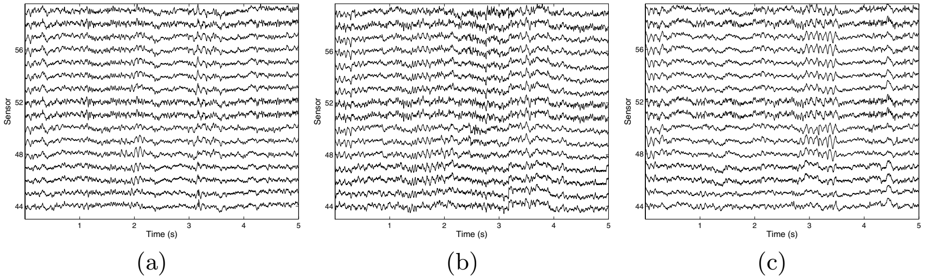


Fig. 4. Five-second segments of the original EEGs recorded by sensors (from Channel 44 to Channel 58). (a) EEGs of the Circle Class after the epoch-finding. (b) EEGs of the Square Class after the epoch-finding. (c) EEGs of the Triangle Class after the epoch-finding.

all of the eigenvalues of $\sum_{i=1}^{P}(\mathbf{R_Z}(i\tau) + \mathbf{R_Z}(i\tau)^T)$, and we extracted the next VEP component:

1. Applied the proposed algorithm to extract the component that corresponded to the $(q+1)$-th largest eigenvalue of $\sum_{i=1}^{P}(\mathbf{R_Z}(i\tau) + \mathbf{R_Z}(i\tau)^T)$;
2. To ensure the extracted component was not the component of artifacts, we calculated its autocorrelation;
3. Since the components of VEPs exhibited time-locked activation to task-related events and those of artifacts did not, the autocorrelations of VEP components had peaks locating at lag 3174, lag 6348, lag 9522, et al., while those of artifacts components did not. By this method, if we found the extracted component was not a component of VEPs, then we discarded it and went back to step 1. It should be noticed that there are many approaches, e.g. [15], that can help us further distinguishing artifacts from evoked potentials.

This loop continued until we extracted three VEP components; each component consisted of 120 trials. Then the epoch-finding was conducted using the Neuroscan toolbox so that the trials corresponding to the same type of figures were gathered into a class (Fig.5 shows the average result of the trials belonging to the same extracted VEP component in each class). Thereby we had three classes, namely the Circle Class, the Square Class and the Triangle Class.

We randomly selected 60 trials of each extracted component as the training set and the remained trials of each extracted component as the test set. For classifi-

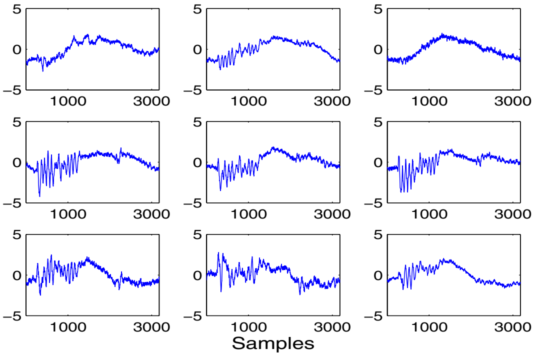


Fig. 5. The averaged trials. The signal in i-th row and j-th column is the average result of the trails belonging to the i-th extracted VEP component of the j-th class ($i, j = 1, 2, 3$).

cation, the feature vectors of each class were constructed as follows: we selected some features from each trial of the first, the second and the third extracted components, respectively, and these features were concatenated orderly to form a feature vector. Here we selected 30 features from the frequency components of each trial according to the MIFS-U algorithm [16], an effective feature selection method based on mutual information. Finally, we used the multi-category SVM as the classifier, and the classification accuracy reached 93.2%.

5 Conclusions

We propose a two-stage algorithm for extracting source signals that satisfy some given temporal structure. The algorithm is suitable to extract the periodic or quasi-periodic source signals, even if the desired source signals have the same period (but they should have different autocorrelation structure). Compared with many widely-used extraction algorithms, the algorithm has better performance, verified by simulations and experiments.

Acknowledgements

The work was supported by the National Basic Research Program of China (Grant No. 2005CB724301) and National Natural Science Foundation of China (Grant No.60375015).

References

1. Cichocki, A., Amari, S.: Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications. John Wiley & Sons, New York (2002)
2. Zhang, Z.-L., Yi, Z.: Robust Extraction of Specific Signals with Temporal Structure. Neurocomputing **69 (7-9)** (2006) 888-893

3. Zhang, Z.-L., Zhang, L.: A Two-Stage Based Approach for Extracting Periodic Signals. Proc. of the 6th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA 2006). LNCS **3889** (2006) 303-310
4. Zhang, Z.-L., Yi, Z.: Extraction of Temporally Correlated Sources with Its Application to Non-invasive Fetal Electrocardiogram Extraction. Neurocomputing **69 (7-9)** (2006) 900-904
5. Lu, W., Rajapakse, J.C.: Approach and Applications of Constrained ICA. IEEE Trans. Neural Networks **16 (1)** (2005) 203-212
6. Barros, A.K., Vigário, R., Jousmäki, V., Ohnishi, N.: Extraction of Event-related Signals from Multichannel Bioelectrical Measurements. IEEE Trans. Biomedical Engineering **47 (5)** (2000) 583-588
7. Barros, A.K., Cichocki, A.: Extraction of Specific Signals with Temporal Structure. Neural Computation **13 (9)** (2001) 1995-2003
8. James, C.J., Gibson, O.J.: Temporally Constrained ICA: An Application to Artifact Rejection in Electromagnetic Brain Signal Analysis. IEEE Trans. Biomedical Engineering **50 (9)** (2003) 1108-1116
9. Cao, J., Murata, N., Amari, S.-I., Cichocki, A., Takeda, T.: A Robust Approach to Independent Component Analysis of Signals with High-Level Noise Measurements. IEEE Trans. Neural Networks **14 (3)** (2003) 631-645
10. Karvanen, J., Koivunen, V.: Blind Separation Methods Based on Pearson System and Its Extensions. Signal Processing **82** (2002) 663-673
11. Paul, J.S., Reddy, M.R., Kumar, V.J.: A Cepstral Transformation Technique for Dissociation of Wide QRS-Type ECG Signals Using DCT. Signal Processing **75** (1999) 29-39
12. Belouchrani, A., Abed-Meraim, K., Cardoso, J.F., Moulines, E.: A Blind Source Separation Technique Using Second-Order Statistics. IEEE Trans. Signal Processing **45 (2)** (1997) 434-444
13. Hyvärinen, A.: Complexity Pursuit: Combining Nongaussianity and Autocorrelations for Signal Separation. Proc. of the 2th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA 2000), pp. 175-180
14. Jafari, M.G., Wang, W., Chambers, J.A., et al.: Sequential Blind Source Separation Based Exclusively on Second Order Statistics Developed for a Class of Periodic Signals. IEEE Trans. Signal Processing **54 (3)** (2006) 1028-1040
15. Shoker, L., Sanei, L., Chambers, J.: Artifact removal from electroencephalograms using a hybrid BSS-SVM algorithm. IEEE Signal Processing Letters **12 (10)** (2005) 721-724
16. Kwak N., Choi C.-H.: Input Feature Selection for Classification Problems. IEEE Trans. Neural Networks **13 (1)** (2002) 143-159

Bispectrum Quantification Analysis of EEG and Artificial Neural Network May Classify Ischemic States

Liyu Huang[1,2], Weirong Wang[1,3], and Sekou Singare[2]

[1] Department of Biomedical Engineering, Xidian University, Xi'an, 710071, China
[2] Institute of Biomedical Engineering, Xi'an Jiaotong University, Xi'an, 710049, China
huangly@mail.xjtu.edu.cn
[3] Department of Medical Instrumentation, Shanhaidan Hospital, Xi'an, 710004. China
owl@mail.xidian.edu.cn

Abstract. This paper examines the relation between the degree of experimentally induced focal ischemia in the left-brain of 24 experimental rats and Higher Order Statistics (HOS) such as the bispectrum and the bicoherence index of scalp EEG recorded at the time of the ischemic event. The aim is to propose the assessment of HOS in non-invasive scalp EEG to facilitate identification and even classification of focal ischemic events in terms of the degree of tissue damage. The latter is achieved by a supervised, multilayer, feed-forward Artificial Neural Network (ANN). The ANN utilizes a back propagation algorithm to classify ischemic states of the brain. The target values used during the training session of the network are the degree of ischemic tissue damage (graded as serious, middle and slight) as assessed by histological and immunhistochemical methods in the brain slice of the experimental animals. The results show that the ANN can correctly identify and classify ischemic events with high precision 91.67% based on HOS measures of scalp EEG obtained during ischemia. These findings may potentially be of great scientific merit, especially due to their possibly very important medical implications: a potential non-invasive method that reliably identifies the presence and the degree of ischemia at the time of its occurrence.

1 Introduction

Cerebral vascular diseases are one of the most important factors influencing morbidity and mortality now [1]. Most of the cerebral vascular diseases can be associated with cerebral ischemia. A noninvasive technique for early detection of brain's ischemic injury is quite needed [2][3], but at present, no objective method to detect and monitor brain's ischemic injury exist in clinical diagnosis.

Since cerebral ischemia can directly affect the brain function, so neuroelectrical signals(such as EEG) analysis seems to be a good choice to look for a correlation between the state of injury and the signals. However, the use of EEG as a measure of estimation of ischemic injury has achieved very limited success, this partly may be attributed to the fact that commonly used signal processing methods is based on the assumption that EEG arises from a linear and stationary process.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 533 – 542, 2006.

The previous works showed that bispectrum analysis of the EEG can yield variables, which might correlate with ischemic cerebral injury. These parameters are sensitive to focal ischemic cerebral injury, such as the maximum magnitude and the WCOB[4]. In this paper, a further method is introduced to detect the extent of cerebral injury.

2 Experiment and Data Acquisition

2.1 Animal Experiment of Focal Brain's Ischemic Injury

Twenty-four adult SD rats weighting 200–350g, either sex, were anesthetized by injecting 2% sodium pentobarbital into abdominal cavity(0.8ml/250g). Left carotid arterys were separated for about 0.8cm length and the blood vessels were tied up at the heart side and incised at another side. A cannula was inserted in the trachea and a thin polyethylene catheter was placed in the common carotid artery, infused physiological saline along the direction of blood flow for 8 min, 18 min and 30 min to cause cerebral mild acute ischemia in different extent. The rate of infusion maintains at 0.2ml/min and the pressure was adjusted at 14 kPa to prevent the blood offering by lateral blood circulation. After surgery, the rat was kept at rest for a few minutes to stabilize it. Environment temperature was maintaining at 28°C ± 3°C.

2.2 EEG Data Collection

After preparation, the rat head was fixed in a stereotaxic frame and the scalp was dissected. Four channels of EEG using subdermal needle electrodes with shielded cable, placed in left-frontal-parietal, left occipital, right frontal-parietal and right occipital areas, labeled lead1, 2, 3, 4, respectively, were recorded by Spectrum32 (CADWELL Lab, USA), with the reference points at the nose and the ground electrodes at the tongue. The EEG data were filtered with a high-pass filter at 0.3*Hz* and a low-pass filter at 70*Hz* and sampled at 200*Hz*, digitized to 12bits. A 50*Hz* notch filter was also employed.

The EEG signals were recorded after graded ischemic injury of 8 min, 18 min and 30 min, respectively. Each record is about 20 s long. The number of the experimental rats in different times of ischemia was arranged as Table 1.

Table 1. Number of rats arranged in the experiment

Ischemia time	Left brain injury
8 min	8
18 min	8
30 min	8

Table 2. States classification in HSP70 and HE

Injury level	Classification rules description
1.0	Injury area was less than 10%
0.5	More than 10% but less 20%
0.0	Injury area was more than 20%

2.3 States Classification of Injury

After the experiment the rats were killed for immunohistochemistry and histopathology study. Some slices were made by cryoultramicrotomy to perform immunohistical chemical experiment (HSP70 expression) and conventional hematoxylin and eosin (HE) staining. State of ischemia was graded as three different levels (serious, middle and slight) by observing the extent from slices of HSP70 and HE staining. The set of non-EEG criteria was given in Table 2. The corresponding EEG was labeled from zero to one in increments of 0.5.

3 Method

3.1 Bispectrum Analysis

The conventional power spectra is useful for studying only the linear mechanisms governing the process since it suppresses phase relations between frequency components[4]. At present, higher-order spectra, especially bispectrum play an important role due to their ability of preserving non-minimum phase information, as well as information due to deviations from Gaussianity and degrees of nonlinearities in time series. Since we expected EEG to have nonlinearities in the generating mechanism, bispectrum analysis of EEG might reveal additional non-Gaussian and nonlinear information due to its certain advantage [5][6].

3.1.1 Definition of Bispectrum

Higher-order spectra are multi-dimensional Fourier transforms of higher-order statistics. Thus, the bispectrum is defined by third-order cumulant or third-moment sequence. Let $X(n)$ be a stationary, discrete, zero-mean random process and its third-order cumulant sequence $c_3^x(\tau_1,\tau_2)$ will be identical to its third-moment sequence:

$$c_3^x(\tau_1,\tau_2)=E\{X(k)X(k+\tau_1)X(k+\tau_2)\} \tag{1}$$

where $E\{\cdot\}$ denotes statistical expectation. The bispectrum $B(\omega_1,\omega_2)$ of $X(n)$ is defined as the two-dimensional(2-D) Fourier transform of $c_3^x(\tau_1,\tau_2)$

$$B(\omega_1,\omega_2)=C_3^x(\omega_1,\omega_2)=\sum_{\tau_1=-\infty}^{\infty}\sum_{\tau_2=-\infty}^{\infty}c_3^x(\tau_1,\tau_2)\exp[-j(\omega_1\tau_1+\omega_2\tau_2)] \tag{2}$$

$$|\omega_1|,\ |\omega_2|\le\pi,\ |\omega_1,\omega_2|\le\pi$$

In general, $B(\omega_1,\omega_2)$ is complex and a sufficient condition for its existence is that $c_3^x(\tau_1,\tau_2)$ is absolutely summable.

3.1.2 TOR Method to Estimate Bispectrum

Consider a real pth order autoregression (AR) process $X(n)$ described by

$$X(n)+\sum_{i=1}^{p}\alpha_i X(n-i)=W(n) \tag{3}$$

where $W(n)$ is a non-Gaussian function with $E\{W(n)\}=0$ and $E\{W^3(n)\}=\beta\neq 0$, α is the AR parameter, and β is the third-order moment of the driving noise. $X(m)$ is independent of $W(n)$ for $m<n$.

Since $W(n)$ is third-order stationary it follows that $X(n)$ is also third-order stationary assuming it is a stable AR model. The third moment of the $X(n)$ is also described as $R(\tau_1,\tau_2)=E\{X(n)X(n+\tau_1)X(n+\tau_2)\}$ and it satisfies the following third-order recursion:

$$R(-k,-l)+\sum_{i=1}^{p}\alpha_i R(i-k,i-l)=\beta\delta(k,l) \tag{4}$$

where $R(\tau_1,\tau_2)$ is the third moment sequence of the AR process and $\delta(k,l)$ is the 2-D unit impulse function. Then, we can estimate the third-order moment $\hat{R}(\tau_1,\tau_2)$ using the conventional indirect bispectrum estimation method [7]. Substituting $R(\tau_1,\tau_2)$ by $\hat{R}(\tau_1,\tau_2)$ in (4), we can obtain the estimated value of bispectrum

$$\widehat{B}(\omega_1,\omega_2)=\widehat{\beta}\ \widehat{H}(\omega_1)\widehat{H}(\omega_2)\widehat{H}^*(\omega_1,\omega_2) \tag{5}$$

or more conveniently the normalized estimate

$$\frac{\widehat{B}(\omega_1,\omega_2)}{\widehat{\beta}}=\widehat{H}(\omega_1)\widehat{H}(\omega_2)\widehat{H}^*(\omega_1,\omega_2) \tag{6}$$

Where

$$H(\omega)=\frac{1}{1+\sum_{n-1}^{p}\alpha_i\exp(-j\omega n)}\qquad|\omega|\le\pi \tag{7}$$

This method to estimate bispectrum is called the *third-order recursion* (TOR) method.

3.1.3 Definition of WOCB

To quantify the diagnosis indicators, Zhang *et al* defined a WCOB[8]. Supposed the bispectrum of point (x, y) is B_{xy}, then the $WCOB(f_{1m}, f_{2m})$ in the bi-frequency plane can be calculated as

$$f_{1m} = \frac{\sum xB_{xy}}{\sum B_{xy}} \qquad f_{2m} = \frac{\sum yB_{xy}}{\sum B_{xy}} \tag{8}$$

The WCOB is a vector with two variables f_{1m} and f_{2m}.

3.1.4 Definition of Bicoherence Index

Bispectrum estimation is useful in detecting and quantifying quadratic phase coupling present between any two-frequency components of a process. A function called the bicoherence index combines two completely different entities namely the bispectrum and the power spectrum of a process and is given by Huber *et al.* in [9] as

$$b(\omega_1, \omega_2) = \frac{B(\omega_1, \omega_2)}{\sqrt{P(\omega_1)P(\omega_2)P(\omega_1, \omega_2)}} \tag{9}$$

where $B(\omega_1, \omega_2)$ and $P(\omega)$ are the bispectrum and the power spectrum of the process, respectively. $b(\omega_1, \omega_2)$ is the bicoherence index at frequencies (ω_1, ω_2).

3.2 Artificial Neural Networks

It is more likely that the focal ischemic states would be differentiated by the bispectrum of the EEGs if we analyze the values of the bicoherence index and WCOB. Among these values and the ischemic states, there may be a certain correlation, which may be difficult to express using analytical methods, but can be captured by a multilayer ANN, as the hidden and output nodes used a logistic sigmoidal activation function to analyze the nonlinearities in the data. In our study, compared with three-layer(one hidden layer) ANNs, the four-layer ANN(two hidden layers) has a certain advantage in estimating the ischemic states(see Table 4). The number of the second hidden units and the optimum number of clusters are determined according to analysis of input and output feature space[10, 11] and pseudo F-statistic(PFS) clustering technique[12]. The optimum result of ANN structure is 12-7-2-1. We build up the network in such a way that each layer is fully connected to the next layer.

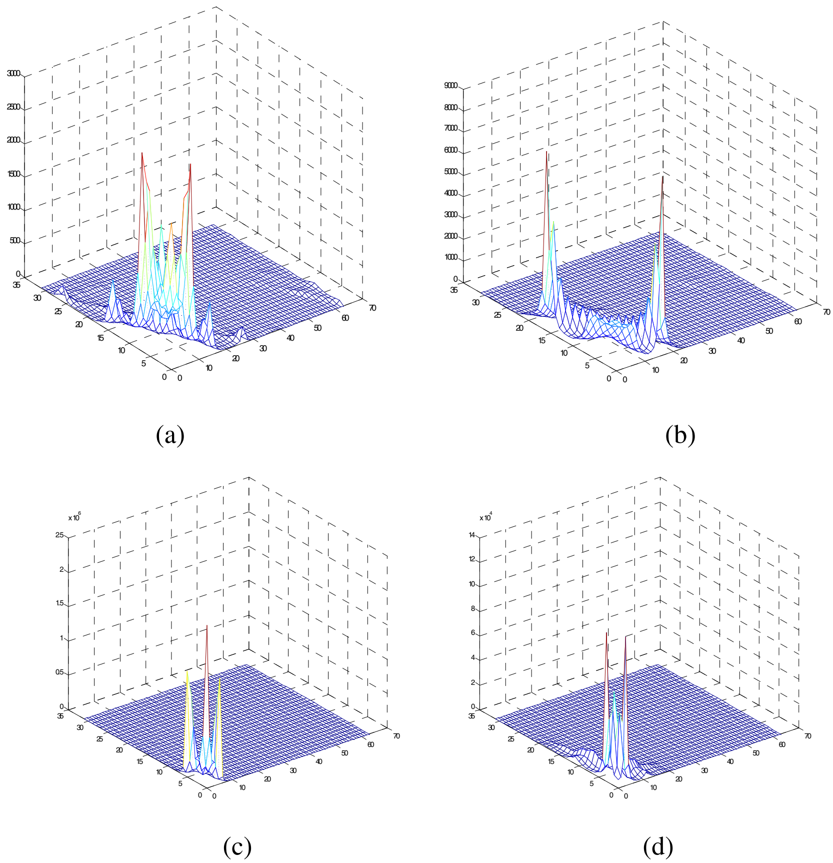
The input vector of the ANN consists of four values of maximum bicoherence index and eight values of f_{1m}, f_{2m} from WCOB, which were extracted from four-channel EEG of each rat. The output will be the estimation result of brain's injury.

Training of the ANN, essentially an adjustment of the weights, was carried out on the training set, using the back-propagation algorithm. Our ANN was trained using ‘leave-one-out’ strategy, because of our small number of sample recording [13].

4 Results

Fig 1 (a)~(d) shows the bispectrum contour maps of a SD rat’s EEG from lead 1 at the four different ischemic stages. From these figures, we can observe that the contour maps become quite different for the left brain at different ischemia stages (different ischemic extent).

Fig 2 shows changes of the brain’s frequency coordinates of WCOB from four-lead EEGs at the different ischemia stages. Before ischemia, the f_1 and f_2 of left and right brain is almost identical. When the ischemia lasts 8 min, all of the frequency



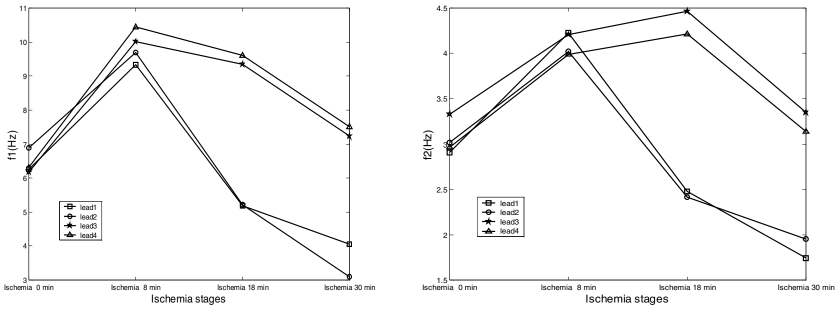
(a)

(b)

(c)

(d)

Fig. 1. Bispectrum contour maps of a SD rat’s EEG from lead 1 at the four different stages, (a) ischemia starting, (b) 8 min, (c) 18 min and (d) 30 min



(a) (b)

Fig. 2. Changes of the frequency coordinates of WCOB during ischemic injury in a typical rat, (a) f_1 (b) f_2

values increase. This may be caused by the emergent response when ischemia is being induced. When the ischemia lasts 18 min, f_1 values decrease. Especially, the values of left brain (ischemic region) decrease more quickly than that of right one. But f_2 values of right brain increase when the left brain's values decrease at this ischemia stage. When the ischemia lasts 30 min, the frequency values of left brain decrease rapidly, while the right brain's values decrease slowly. The values of left brain are not only less than that of right brain, but also far less than that of normal state (ischemia 0 min).

Fig. 3 displays changes of maximum bicoherence index at different ischemia stages. When the ischemia is induced, three of the four values decrease. With the continuance of ischemia, all indexes increase. But the values of left brain increase quickly, the right brain's values increase just a little.

Although these parameters seem to be rather sensitive to ischemic extent, it is difficult to use them as an accurate measure for quantitative discrimination of ischemic states. ANN can help us to capture the certain nonlinear correlation between these parameters and the ischemic extent.

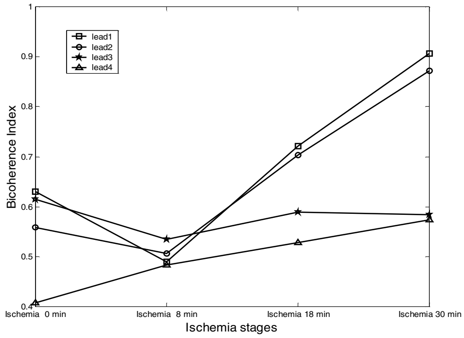


Fig. 3. Changes of the maximum bicoherennce indexes during ischemic injury in a typical rat

According to the rule of injury state classification, the results of testing our proposed scheme are shown in Table 3. In total, one rat with slight ischemic state was misclassified as middle ischemia, and one rat with middle ischemic state was misclassified as slight one. For injury extent assessment, the average accuracy is 91.67%. We tested our scheme using different ANNs, for four-layer (12-7-2-1) and three-layer (12- 4-1) ANNs, the accuracy for the injury extent prediction is 91.67 and 83.33%, respectively. The comparison of the performances is shown in Table 4.

Table 3. Testing results of ischemic extent(12-7-2-1)

extent of ischemia induced	Number of rats	Misclassified rate for injury extent	Accuracy
slight	8	1/8	87.5%
middle	8	1/8	87.5 %
serious	8	0/8	100 %

Table 4. Comparison of performances of different ANN employed

Types of ANN employed	Accuracy for the extent prediction
Four-layer ANN (12-7-2-1)	91.67 %
Three-layer ANN (12-4-1)	83.33 %

5 Discussion

The previous study shows [4] that the contour maps of brain's bispectrum at the different ischemic stages may give more comprehensive information in another way. The power spectra cannot show the obvious difference among the different ischemia stages and distinguish the ischemic region; the visible changes of the EEG rhythm parameters δ , θ , α and β are also not clear.

Bispectrum is based on the third-order statistics which preserves phase information present in a signal, unlike the power spectrum that is phase blind. The phase of a signal is particularly critical in analyzing nonlinear systems where sinusoidal components of distinct frequencies could interact nonlinearly to produce one or more sinusoidal components at sum and difference frequencies [5,14,15]. EEG, being generated by a nonlinear system, would be expected to have many such sinusoidal components produced due to the nonlinearity in the system. The third-order statistics, therefore, help in identifying those components. In this paper, we propose a new approach to early detect ischemic extent using bispectrum quantification analysis method. The results show that the approach is a new potential way to assess cerebral ischemic injury. Our studies further indicate that, in most cases, the EEG contains sufficient information to estimate brain's ischemia, the key is whether or not the method used is suited to the nature of the EEG signal properly.

Early quantitative diagnosis of cerebral injury and prognosis for neurological recovery are a complicated concept and it is difficult to be accurately evaluated by a single parameter or single method. Combination of different methods, especially nonlinear methods may be a potential trend.

During the on-line application of our system, the recorded EEG and other parameters can also be stored in the specific database for updating. Thus, every certain period, the ANN is retrained off-line using the newly updated specific database, and then the new weights are sent to the trained ANN to update its weights. In so doing, the system can keep 'dynamic update' during real application.

When the number of samples is very large, it would be reasonable to partition the data into a training set and a test set. But owing to the restricted experimental conditions, we have only 24 recordings from 24 SD rats for training and testing the ANN. A good way to tackle such a dilemma would be to train the ANN on samples from *n–1* rats and test on samples from the remaining one. This process is repeated *n* times and each time a different rat is left behind. That is so-called 'drop-one-rat' method, or 'leave-one-out' method [13], it is a standard method to evaluating classification systems, especially in the case of small samples for training and test. As the training and test samples belong to different rats, there is not any bias in the results.

To supply the criterion for distinguishing the states of ischemic injury, we analyzed the cellular expression of HSP70 and conventional HE staining in the animal brain. The HE stained and HSP test show that the left brain is lightly insulted, and the right brain is normal. No HSP70 proteins were found in the normal brain tissue. This is identical to our analysis result. These methods can successfully verify our ischemic experiment model and our new approach.

Although the results of this initial study are significant, more animal experiments and clinical studies need to be performed to test the effectiveness of the method.

Acknowledgment

This work was supported by China Postdoctoral Science Foundation and National Natural Science Foundation of China under grant NO. 60371023.

References

1. Cerutii, S., Bersani, V., Carrara, A., (eds.): Analysis of visual evoked potentials through Wiener filtering applied to a small number of sweeps. J. Biomed. Eng. 9 (1987) 3–12
2. Thakor, N. V.: Adaptive filtering of evoked potentials. IEEE Trans. Biomed. Eng., 34 (1987)6–12
3. Bertrand, O., Bohorquez, J., Pernier, J.: Time-frequency digital filtering based on invertible wavelet transform: An application to evoked potentials. IEEE Trans. Biomed. Eng. 41(1994) 77– 88
4. Zhang, J., Zheng C., Xie A.: Bispectrum Analysis of Focal Ischemic Cerebral EEG Signal Using Third-Order Recursion Method. IEEE Trans. Biomed. Eng. 47(2000) 352–359

5. Raghuveer, M. R., Nikias, C. L.: Bispectrum estimation: A parametric approach. IEEE Trans. Acoust, Speech, Signal Processing. 33 (1985) 1213–1230
6. Nikias, C. L., Raghuveer, M. R.: Bispectrum estimation: A digital signal processing framework. Proc. IEEE, 75 (1987) 869–891
7. Nikias, C. L.: Higher-Order Spectra Analysis: A Nonlinear Signal Processing Framework. Englewood Cliffs, NJ: Prentice-Hall (1993)
8. Zhang J., Zheng C., Jiang D., (eds.): Bispectrum analysis of focal ischemic cerebral EEG signal. Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 20 (1998) 2023–2026
9. Huber, P. J., Kleiner, B., Gasser, T., Dumermuth, G.: Statistical methods for investigating phase relations in stationary stochastic processes. IEEE Trans. Audio Electroacoust., AU-19 (1971) 78–86
10. Lippmann, R. P.: An Introduction to Computing with Neural nets. IEEE ASSP Magazine, April (1987) 4–22
11. Mirchandami, G., Cao, W.: On hidden nodes for neural nets. IEEE Trans. on Circuits and System, 36 (1989) 661–664
12. Vogel, M. A., Wong, A. K. C.: PFS clustering method. IEEE Trans. on Pattern Anal. Mach. Intell. 1 (1979) 237–245
13. Fukunaga, K.: Introduction to Statistical Pattern Recognition. 2nd edition, Academic, San Diego, CA (1990)
14. Giannakis, G.B., Mendel, J.M., Cumulant-based order determination of Non-Gaussian ARMA models. IEEE Trans Acoust, Speech, Signal Processing, 1990; 38: 1411–1423.
15. Giannakis, G.B., Mendel, J.M., Identification of nonminimum phase systems using higher-order statistics. IEEE Trans Acoust, Speech, Signal Processing, 1989; 37: 360–377.

An Adaptive Beamforming by a Generalized Unstructured Neural Network

Askin Demirkol, Levent Acar[1], and Robert S. Woodley[2]

[1] Department of Electrical and Computer Engineering, University of Missouri-Rolla,Missouri US
[2] 21st Century Systems, Inc.Fort Leonard Wood, Missouri US

Abstract. In this paper, an adaptive array beamforming by an unstructured neural network based on the mathematics of holographic storage is presented. This work is inspired by similarities between brain waves and the wave propagation and subsequent interference patterns seen in holograms. Then the mathematics to produce a general mathematical description of the holographic process is analyzed. From this analysis it is shown that how the holographic process can be used as an associative memory network. Additionally, the process may also be used a regular feed-forward network. The most striking aspect of these network is that, using the holographic process, the apriori knowledge of the system may be better utilized to tailor the neural network for an adaptive beamforming problem. This aspect, makes this neural network formation process particularly useful for the beamforming.

Keywords: holographic processing, wave propagation, Green's functions, radial basis functions, feed-forward neural network, adaptive beamforming.

1 Introduction

Here a hologram formation process will be analyzed by the distributed signal processing principles. We will study on how we may use the knowledge gained from holograms to construct the adaptive beamforming radial basis network. A hologram is formed when momochromatic, coherent light is reflected off an object, then interfered with by another monochromatic, coherent reference beam[1]. Since the beams are monochromatic, they can be represented in rotating phasor form

$$u(x,t) = \Re\{A(x)e^{j\phi(x)}e^{j2\pi ft}\} \tag{1}$$

where x is a position, and f is the frequency. This monochromatic wave must satisfy the Maxwell equation[2],

$$\nabla^2 u - \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = 0 \tag{2}$$

Since the time dependence is known a priori[3], the complex phasor function

$$U(x) = A(x)e^{j\phi(x)} \tag{3}$$

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 543–552, 2006.

may be used. Equation 3 must then satisfy the Helmholtz equation[3],

$$(\nabla^2 + k^2)U = 0 \tag{4}$$

where $k = 2\pi v/c = 2\pi\lambda$ is the wave number. Equation 4 is also known as the reduced wave equation, and has a known solution using Green's functions[4]. Let G be defined such that

$$LG = \delta(x_\alpha - x_\beta) \tag{5}$$

For a system with operator L

$$LU = h \tag{6}$$

for all x in a volume V. Multiplying equation 6 by G and Equation 5 by $U(x_\beta)$, then integrating and subtracting the equation produce

$$\int_V (U(x_\beta)LG(x_\alpha, x_\beta) - G(x_\alpha, x_\beta)LU(x_\alpha))dV = U(x_\beta - \int_V G(x_\alpha, x_\beta)h(x_\alpha)dV \tag{7}$$

where the α-plane is the object plane and theβ-plane is the recording plane. Equation 7 is in a general mathematical form, the specifics of the hologram problem reduces the complexity. The operator L has the form

$$LU = (\nabla^2 + k^2)U \tag{8}$$

Therefore, Equation 7 may be simplified and rewritten as

$$\int_V U(x_\beta)(\nabla^2)G(x_\alpha, x_\beta)dV - \int_V G(x_\alpha, x_\beta)(\nabla^2)U(x_\alpha)dV = U(x_\beta) \tag{9}$$

However, the left side of Equation 9 can be evaluated as

$$\int_V (U(x_\beta(\nabla^2)G(x_\alpha, x_\beta) - G(x_\alpha, x_\beta)(\nabla^2)U(x_\alpha)dV = \int_{\partial V} (U\frac{\partial G}{\partial n} - G\frac{\partial U}{\partial n})dS \tag{10}$$

where n is the normal to the surface ∂V of the volume V. With proper choice of the Green's function[3] the right hand side of Equation 10 reduces to

$$\int_V (U(x_\beta(\nabla^2)G(x_\alpha, x_\beta) - G(x_\alpha, x_\beta)(\nabla^2)U(x_\alpha)dV = \int_{\partial V} U\frac{\partial G}{\partial n}dS \tag{11}$$

A reference wave R, is now added to the propagated wave to store multiple images at various angles.

$$I(x_\beta) = |R(x_\beta) + U(x_\beta)|^2 \tag{12}$$

The propagated wave equation is actually an integral transform equation[5]. The kernel of the equation is

$$K_{\alpha-\beta}(x_\alpha, x_\beta) = \frac{\partial G(x_\alpha, x_\beta)}{\partial n} \tag{13}$$

The formation process may now be written in a general mathematical form. Let $U_\alpha(x_\alpha)$ be the signal from the object. This signal propagates to a new location x_β such that

$$U_\beta(x_\beta) = \int_{S_\alpha} U_\alpha(x_\alpha) K_{\alpha-\beta}(x_\alpha, x_\beta) dx_\alpha \tag{14}$$

where S_α is the α-plane. This solution is for only some operators[6]. The mathematical description of L such that U_β in Equation 14 is a solution is currently being investigated. An important result occurs, when $\delta(x_\alpha - x_R)$ is the boundary condition, where δ is the Dirac delta function and x_R is a reference point on the α-plane. The resulting signal at a point x_γ is

$$U_\gamma(x_\gamma) = \int_{S_\alpha} \delta(x_\alpha - x_R) K_{\alpha-\gamma}(x_\alpha, x_\gamma) dx_\alpha = K_{\alpha-\gamma}(x_R, x_\gamma) \tag{15}$$

Similarly, at a point x_β the wave would be

$$U_\beta(x_\beta) = K_{\alpha-\beta}(x_R, x_\beta) \tag{16}$$

However, if we let the β-plane be the boundary, and observe the signal at the γ-plane, then we get

$$U_\gamma(x_\gamma) = \int_{S_\alpha} \delta(x_\alpha - x_R) \int_{S_\beta} K_{\alpha-\beta}(x_\alpha, x_\beta) K_{\alpha-\gamma}(x_\alpha, x_\gamma) dx_\beta dx_\alpha \tag{17}$$

From Equations 16 and 17,

$$K_{\alpha-\gamma}(x_\alpha, x_\gamma) = \int_{S_\beta} K_{\alpha-\beta}(x_\alpha, x_\beta) K_{\beta-\gamma}(x_\beta, x_\gamma) dx_\beta \tag{18}$$

The general mathematical description of the recorded wave is

$$\psi(x_\beta) = U_\beta(x_\beta) + R(x_\beta) \tag{19}$$

The recorded information is actually the norm of ψ. Let $I(x_\beta)$ represent the intensity stored at a location x_β, then

$$I(x_\beta) =< \psi(x_\beta, \psi(x_\beta > \tag{20}$$

where $< .,. >$ is the inner product. For the case of the hologram, the norm is

$$I(x_\beta) = (\psi(x_\beta)^* \psi(x_\beta))^{1/2} \tag{21}$$

where $*$ designates the complex conjugate transpose.

2 Hologram Reconstruction

If the magnitude and phase information are stored, an inversion process may be used to recover $U_\alpha(x_\alpha)$[7]. Let

$$Z(\xi) = \int_{S_\beta} H(\xi, x_\beta) U_\beta(x_\beta) dx_\beta \tag{22}$$

be an invertible transform. then,

$$\begin{aligned} Z(\xi) &= \int_{S_\beta} H(\xi, x_\beta) \int_{S_\alpha} K(x_\alpha, x_\beta) U_\alpha(x_\alpha) dx_\alpha dx_\beta \\ &= \int_{S_\beta} \int_{S_\alpha} H(\xi, x_\beta) K(x_\alpha, x_\beta) U_\alpha(x_\alpha) dx_\alpha dx_\beta \end{aligned} \tag{23}$$

At this point, a restriction needs to be placed upon the kernel K. The kernel must be able to be written as

$$K(x_\alpha, x_\beta) = K(x_\beta - x_\alpha) \tag{24}$$

With this restriction, the function for $U_\beta(x_\beta)$ becomes a convolution integral.

$$\begin{aligned} Z(\xi) &= \int_{S_\beta} H(\xi, x_\beta) \int_{S_\alpha} K(x_\beta - x_\alpha) U_\alpha(x_\alpha) dx_\alpha dx_\beta \\ &= \int_{S_\beta} H(\xi, x_\beta) K(x_\beta) dx_\beta \int_{S_\alpha} H(\xi, x_\alpha) U_\alpha(x_\alpha) dx_\alpha \end{aligned} \tag{25}$$

if,

$$\int_{S_\beta} H(\xi, x_\beta) K(x_\beta - x_\alpha) dx_\beta = H(\xi, x_\alpha) \int_{S_\beta} H(\xi, x_\beta) K(x_\beta) dx_\beta \tag{26}$$

Let $x_c = x_\beta - x_\alpha$, then the left hand side of Equation 26 becomes

$$\int_{S_\beta} H(\xi, x_\beta) K(x_\beta - x_\alpha) dx_\beta = \int_{S_\beta} H(\xi, x_c + x_\alpha) K(x_c) dx_c \tag{27}$$

or,

$$\int_{S_\beta} H(\xi, x_\beta) K(x_\beta - x_\alpha) dx_\beta = \int_{S_\beta} H(\xi, x_\beta + x_\alpha) K(x_\beta) dx_\beta \tag{28}$$

From, Equation 28 and Equation 26

$$\int_{S_\beta} H(\xi, x_\beta + x_\alpha) K(x_\beta) dx_\beta = \int_{S_\beta} H(\xi, x_\alpha) H(\xi, x_\beta) K(x_\beta) dx_\beta \tag{29}$$

or,

$$H(\xi, x_\beta + x_\alpha) = H(\xi, x_\alpha) H(\xi, x_\beta) \tag{30}$$

Equation 30 is valid for most of the transform such as the Fourier transform[5]. To complete the inversion of U_β, rearranging Equation 23, combining it with Equation 25, and letting H^{-1} be the inverse operator of H, then

$$U_\alpha(x_\alpha) = \int_{S_\beta} \int H^{-1}(x_\alpha, \xi) (\int_{S_\beta} H(\xi, x_\beta) K(x_\beta) dx_\beta)^{-1} H(\xi, x_\beta) d\xi U_\beta(x_\beta) dx_\beta \tag{31}$$

under this condition we do not require the use of a reference signal, but we require the storage of the magnitude and the phase of the signal.

The reconstruction begins by multiplying $I(x_\beta)$ and $R(x_\beta)$. The new signal is then propagated to the γ-plane[2], such that

$$\begin{aligned} U_\gamma(x_\gamma) &= \int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) I(x_\beta) R(x_\beta) dx_\beta \\ &= \int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) < U_\beta(x_\beta), U_\beta(x_\beta) > R(x_\beta) dx_\beta \\ &+ \int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) < R(x_\beta), R(x_\beta) > R(x_\beta) dx_\beta \\ &+ \int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) < U_\beta(x_\beta), R(x_\beta) > R(x_\beta) dx_\beta \\ &+ \int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) < U_\beta(x_\beta), R(x_\beta) >^* R(x_\beta) dx_\beta \end{aligned} \tag{32}$$

The first two inner product terms of Equation 32 are the squared norms of the propagated wave and the reference wave, respectively. The third inner product term will be the conjugate image, while the last inner product term will be the recovered image after filtering[2]. Equation 32 is rewritten using the inner product for holograms as

$$\begin{aligned} U_\gamma(x_\gamma) &= \int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) ||U_\beta(x_\beta)||^2 R(x_\beta) dx_\beta \\ &+ \int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) ||R(x_\beta)||^2 R(x_\beta) dx_\beta \\ &+ \int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) U_\beta(x_\beta)^* R(x_\beta) R(x_\beta) dx_\beta \\ &+ \int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) U_\beta(x_\beta) R(x_\beta)^* R(x_\beta) dx_\beta \end{aligned} \tag{33}$$

In the first term of Equation 33, we see that the two norm terms will only change the magnitude of the resulting signal. Therefore, the first term in Equation 33 represents just a scaled version of the propagated reference wave at the same angle of the reference wave. The second term of Equation 33 has the reference wave squared. The resulting image will then be at twice the angle of the reference wave. The last term of Equation 33 has $R(x_\beta)^* R(x_\beta)$ which is simply the magnitude squared of the reference wave. The last term, therefore, becomes

$$R_0^2 \int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) U_\beta(x_\beta) dx_\beta \tag{34}$$

By selecting the angle of the reference R properly, we may filter out the first two terms of Equation 33. Figure 1 shows how process. By placing a spatial filter centered on-axis that is the same size as the original image, the third term of Equation 33 is extracted as

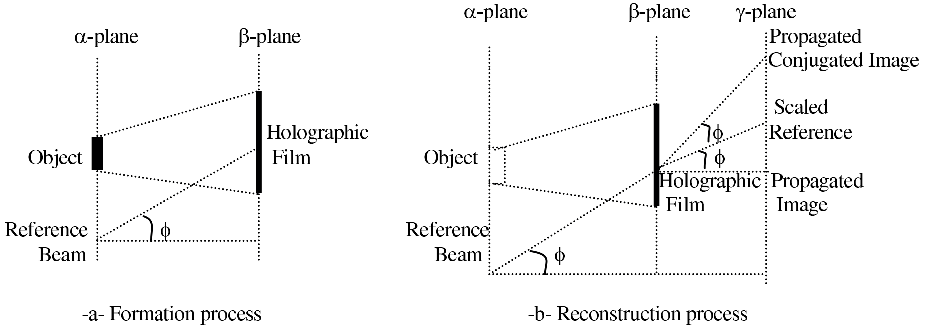


Fig. 1. Schematics of the hologram processes

$$U_{filtered}(x_\gamma) = R_0^2 \int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) U_\beta(x_\beta) dx_\beta \tag{35}$$

Equation 14 may be combined with Equation 35 to produce the final result. Such that,

$$\begin{aligned} U_{filtered}(x_\gamma) &= R_0^2 \int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) \int_{S_\alpha} U_\alpha(x_\alpha) K_{\alpha-\beta}(x_\alpha, x_\beta) dx_\alpha dx_\beta \\ &= R_0^2 \int_{S_\alpha} [\int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) K_{\alpha-\beta}(x_\alpha, x_\beta) dx_\beta] U_\alpha(x_\alpha) dx_\alpha \end{aligned} \tag{36}$$

The bracketed term in Equation 36 has already been shown to be $K_{\alpha-\gamma}(x_\alpha, x_\gamma)$ by Equation 18. Therefore, it is possible to recover $U_\alpha(x_\alpha)$ by inverting the equation for $U_{filtered}(x_\gamma)$. To show multiple images are recovered from the same storage media, we take the sum of the inner products of the multiple sources as

$$I(x_\beta) =< U_1 + R_1, U_1 + R_1 > + < U_2 + R_2, U_2 + R_2 > \tag{37}$$

where U_1 and U_2 are the propagated waves and R_1 and R_2 are the reference waves with different angles of incident. If the signal from U_2 is desired, $I(x_\beta$ is multiplied by the reference wave R_2, such that

$$\begin{aligned} U_\gamma(x_\gamma) = &\int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) < U_1 + R_1, U_1 + R_1 > R_2 dx_\beta \\ &+ \int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) < U_2 + R_2, U_2 + R_2 > R_2 dx_\beta \end{aligned} \tag{38}$$

The second term of Equation 38 will produce the same case as in Equation 33. The first term, however, becomes

$$\begin{aligned}\int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) < \psi, \psi > R_2 dx_\beta = &\int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma)(||U_1||^2 + ||R_1||^2) R_2 dx_\beta \\ &+ \int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) < U_1, R_1 > R_2 dx_\beta \\ &+ \int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) < U_1 + R_1 >^* R_2 dx_\beta\end{aligned} \tag{39}$$

where $\psi = U_1 + R_1$. The only term in Equation 39 that will not shift the image by at least the angle of R_2 is $(\int_{S_\beta} K_{\beta-\gamma}(x_\beta, x_\gamma) < U_1 + R_1 >^* R_2 dx_\beta)$. The angles of R_1 and R_2 will cancel in this case. But by picking the angles as multiplies of a minimum angle ϕ we get $|\angle R_2 - \angle R_1| > \phi$. Therefore, the resulting image will be off-axis and will be filtered out by on-axis filter. The only image passing through the filter will then be the desired image from U_2.

3 Application of the Hologram Process to Radar Beamforming

We will now show the hologram process can be used to create unstructured neural networks. We will show that a feed-forward network which is a single hidden layer radial basis function network is a subset of the general form derived from holograms. And we will examine an adaptation of this network to adaptive beamforming problem.

3.1 Adaptive Beamforming by Radial Basis Function Network

If a linear array system is considered with m identical isotropic sensors, where the sensor separation is D as shown in Figure 2, the signal $x_i(n)$ received at the ith array sensor is given by

$$x_i(n) = \sum_{k=1}^{p} s_k e^{j\omega_0 (i-l)\tau_k} \tag{40}$$

for $i = 1, ..., m$, where $\omega_0 = 2\pi f_c$, $\tau_k = (D/c) sin\theta_k$, the kth signal comes from the directions of arrival θ_k for $k = 1, ..., p$, the carrier frequency and speed of propagation are f_c and c, and the source signal $s_k(n)$ is independent of $s_l(n)$ with $\theta_k \neq \theta_l$ for $k \neq l$. From Equation 40, we have a vector form to express the obtained sensor data

$$\mathbf{x}(n) = \mathbf{A}(\theta)\mathbf{s}(n) \tag{41}$$

where $\mathbf{x}(n) = [x_1(n), ..., x_m(n)]^T$, the steering matrix $\mathbf{A}(\theta)$ and signal vector $\mathbf{s}(n)$ are defined by $\mathbf{A}(\theta) = [\mathbf{a}(\theta_1), ..., \mathbf{a}(\theta_p)$, $\mathbf{s}(n) = [s_1(n), ..., s_p(n)]^T$, and the array response vector $\mathbf{a}(\theta_k)$ is given by $\mathbf{a}(\theta_k) = [1, e^{j\omega_0 \tau_k}, ..., e^{j\omega_0 (m-l)\tau_k}]^T$. Thus the array output $y(n)$ can be written as

$$y(n) = \mathbf{w}^H \mathbf{x}(n) \tag{42}$$

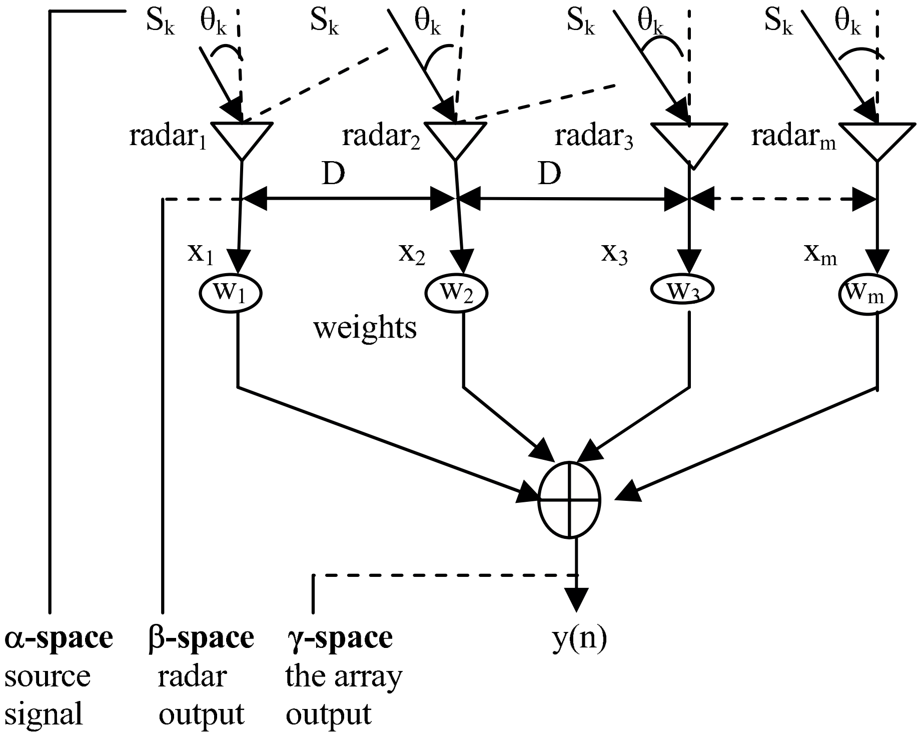


Fig. 2. Mapping processes among Source, Radar and Array output spaces

where $\mathbf{w} = [w_1, ..., w_m]^T$ is the weight vector, and H denotes complex conjugation transpose. The array output in Equation 42 can be written as

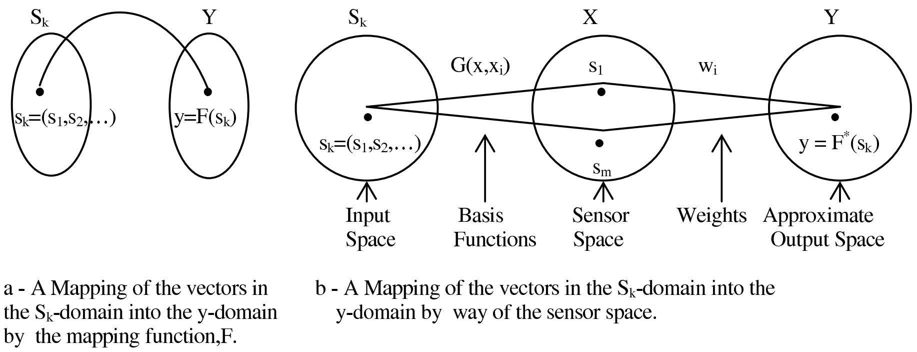
$$y = \sum_{i=1}^{m} \mathbf{w}_i^H \mathbf{x}_i \tag{43}$$

A linear array beamforming process may be viewed as a mapping from one space to another. An unknown system y, maps the input vector $\mathbf{S_k}$ to the output y as shown in Figure 3a. If an unknown system F, maps the input vector $\mathbf{S}_k(n)$ to the output y, such that $y = F(\mathbf{S}_k(n))$ as shown in Figure 3a. Let adapt this to Figure 2. A linear array beamforming can approximate F by first creating the signal $x_i(n)$ received at the ith array sensor is given by Equation 40. The output in Equation 43 can then be written as

$$y = F^*(\mathbf{x}) = \sum_{i=1}^{m} w_i G(\mathbf{x}, \mathbf{x}_i) \tag{44}$$

where F^* is the approximation of F, w_i for $i = 1, ..., m$ are the weights, and $G(\mathbf{x}, \mathbf{x}_i)$ is the Green's function (basis function). In matrix notation

$$\mathbf{F}^* = \mathbf{GW} \tag{45}$$



a - A Mapping of the vectors in the S$_k$-domain into the y-domain by the mapping function,F.

b - A Mapping of the vectors in the S$_k$-domain into the y-domain by way of the sensor space.

Fig. 3. Mapping processes among Input, Feature and Output spaces

3.2 Generalized Radial Basis Beamforming Derived from Holograms

A generalized radial basis adaptive beamforming network can be naturally developed from the mathematics of holograms. If we first consider the object plane as the input space, then each location in the object plane represents an input vector of the **x**-space. Therefore, if we let the input to the hologram be

$$U_\alpha(x_\alpha) = \delta(x_\alpha) \tag{46}$$

then we have only the vector **x** as input into the network. The β-plane then becomes the feature space, where

$$U_\beta(x_\beta) = K_{\alpha-\beta}(x_\alpha, x_\beta) \tag{47}$$

The filtered output of the hologram is then

$$U_{filtered}(x_\gamma) = \int_{S_\beta} R_0^2 K_{\beta-\gamma}(x_\beta, x_\gamma) K_{\alpha-\beta}(x_\alpha, x_\beta) dx_\beta \tag{48}$$

The output location x_γ is arbitrary, therefore the generalized neural network approximating the function F may be written as

$$F^*(x_\alpha) = \int_{S_\beta} R_0^2 K_{\beta-\gamma}(x_\beta, x_\gamma) K_{\alpha-\beta}(x_\alpha, x_\beta) dx_\beta \tag{49}$$

By this way, Figure 3b can be seen as the feature space for the generalized network. Creating a radial basis beamforming function neural network from the generalized form is straight forward. First, the feature space of the generalized network is continuous, so by making it discrete we have the same type feature space as the radial basis network. The equation for the network is then

$$F^*(x_\alpha) = \sum_{i=1}^{N} R_0^2 K_{\beta-\gamma}(x_{\beta_i}, x_\gamma) K_{\alpha-\beta}(x_\alpha, x_{\beta_i}) dx_\beta \tag{50}$$

Comparing this to the radial basis form, we see that

$$K_{\alpha-\beta}(x_\alpha, x_{\beta_i}) = G(x_\alpha, x_{\beta_i}) = \mathbf{G}(\mathbf{x}, \mathbf{x}_i) \tag{51}$$

and

$$w_i = R_0^2 K_{\beta-\gamma}(x_{\beta_i}, x_\gamma) \tag{52}$$

where x_γ is arbitrary. The main benefit of the generalized derivation compared to the derivation found in[9] is that more information about the system may be utilized. The generalized network uses the kernel K to perform the mapping. However, K is generated from the differential equation Lu. So, by changing the kernel, it should be possible to create different classes of neural networks besides the radial basis network.

4 Conclusion

In this study, we presented a radial basis adaptive beamforming network based upon the mathematical description of holographic storage. The network is unstructured in that the hologram process can produce the same results as a number of different neural network structures within a single context. The only change needed for the hologram process is the specific kernel. We concluded by showing how the hologram process is a superset of radial basis beamforming networks. The most important feature of the process is the kernel. Any network may be created if the kernel is known. Furthermore, we have shown that neural networks can be grounded in physical laws. Once the kernel is known, the organization of network is known. This then gives the designer another tool to use in generating a better system.

References

1. Jenkins,F.A.,White,H.E.: Fundamentals of Optics, (1976).
2. DeVelis,J.B., Reynolds,G.O.: Theory and Applications of Holography. (1967).
3. Goodman,J.W.: Introduction to Fourier Optics. (1968).
4. Roach,G.F.: Green's Functions, 2*nd* edition.(1970).
5. Arfken,G.: Mathematical Methods for Physicists. (1970).
6. Debnath,L., Mikusinski,P.: Introduction to Hilbert Spaces with Appl.(1990).
7. Schneider,W.A.: "Integral formulation for migration in two and three dimensions", Geophysic, vol.43, no.1, (1978), pp.49-76.
8. Zurada,J.M.: Introduction to Artificial Neural Systems. (1995).
9. Haykin,S.: Neural Networks: A Comprhensive Foundation. (1999).

Application of Improved Kohonen SOFM Neural Network to Radar Signal Sorting

Chuang Zhao and Yongjun Zhao

ZhengZhou Information Science and Technology Institute,
ZhengZhou, China, 450002
rushzhao@163.com

Abstract. Kohonen neural network is capable of self-organizing and recognizing clustering center, which is used in many artificial intelligence (AI) fields. One electronic support measures (ESM) system must sort the received radar pulses to cells with same features by pulse parameters, such as radio frequency (RF), angle of arrival (AOA), pulse width (PW), Pulse Repetition Interval(PRI), etc. Kohonen SOFM algorithm is one valid method for clustering, which can be used to accomplish such radar pulses sorting. Considering the variety character of pulses parameters which is the character of modern radar system, a new definition of "distance" in the SOFM neural net is proposed in this paper, which decreases the effect of large variety range of special parameter among them. This paper employs the "distance" to improve the clustering capability in such special environments. The computer simulation shows the validity of these improvements.

1 Introduction

In the dense electromagnetic environments encountered in modern warfare, the ESM receiver may receive large number of pulses shown as pulse stream from different emitters. In order to identify individual emitters, their pulse trains must be segregated. The ESM receiver is a passive radar receiver, picks up the pulses transmitted by various radars emitters and measures their parameters, which are angle of arrival (AOA), radio frequency (RF), pulse width (PW), pulse amplitude (PA) and time of arrival (TOA). The measured parameters of every intercepted pulse are encoded in digital format called the pulse descriptor vector (PDV). And the deinterleaver sorts the PDVs and forms pulse cells, each containing a set of PDVs assumed to belong to the same radar emitter. Then, other parameter is generated, which is pulse repetition interval (PRI), and its definition is:

$$PRI_i = TOA_i - TOA_{i-1} \tag{1}$$

Generally, deinterleaving algorithms are classified on the basis of whether they use the parameters of more than one pulse such as the pulse repetition interval (PRI), or they use the parameters of a single pulse such as AOA, RF, and PW [5]. The multiple

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 553 – 559, 2006.

parameters deinterleaving algorithm will improve the reliability and the processing speed, compared with the former algorithm.

In the ESM system, many methods are applied to sort the pulse train. Such methods can solve the problems that the received signal pulses are not much polluted by the performance of the receiver or by the noise. Frankly speaking, the pulses in each radar cell may not be all the ones which are transmitted by the outer radar emitter. One or more pulses may be lost by the receiver, which requires the flexibility of the deinterleaver for sorting. And in ESM system, the period of process is strictly limited, which demands the method with the character of rapidity. Considering all above, the improved artificial neural net, Self-Organizing Feature Map net is used to the field of radar pulse sorting.

SOFM net as one method without human participant can be use to the ESM system, which has the character of topology order preserving and can form clustering by feature. Then this paper uses improved SOFM net to sort the received radar pulse train.

2 Structure of SOFM Net

2.1 SOFM Net Principle

SOFM method, also called Kohonen method, was first proposed by Kohonen in 1982, which is an artificial neural net without teacher. It can study automatically from the environments, and is applied widely to many fields, such as voice recognition, image compress, robot control, etc. Such net is based on the physiology and brain science. According to Kohonen, the human neural net is divided into different parts to respond to different input patterns, and all of these are done by itself. It has the ability of lateral association, and its output nodes are distributed as two-dimension array. The output nodes are connected with the others, and affect each other. All the output nodes in one near zone have resembling outputs, and the distribution of such clustering is similar with the input pattern.

2.2 Structure of SOFM Net

The Kohonen SOFM net consists of two layers, which are the input layer and the output layer. Every nerve cell of the input layer is connected with every nerve cell of the output layers. The cells of output layer are arranged by two dimensions structure. Each corresponds to one input pattern. The Structure is shown as figure 1. The process of competition in the output layer is described as follows: the cells of the near zone N_c of one "win" cell c will be excited to different degree, while the cells out of the zone will be restrained. Nc(t) is a function varied by t. It will decrease with the increase of t until only one or one group nerve cell is left, which delegates the property of the input class pattern.

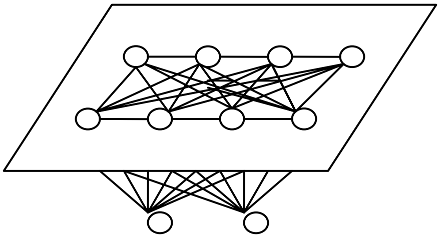


Fig. 1. The structure of SOFM

3 Improved SOFM Algorithm

The improved algorithm is based on the character of radar pulse parameters in ESM environments. With the radar system becoming more and more complex, many radar signal parameters are not single, or stable, which causes the difficulty of the ESM system. For example, the frequency of the frequency-jump radar can change in the range of 5~10% around the central value; and the frequency of frequency-agile radar system can change in the range of 30% around the central value; and the PRI-change radar system, such as stagger PRI radar system(it means the PRI will stagger between the 5% around the central value), etc.

This paper first analyses the pulse signal character of different radar systems, and improves the Kohonen SOFM net to accomplish the process of radar pulse sorting work.

3.1 Revised Definition of "Distance"

When one or more parameters in the training patterns vary in a large range, it can cause the SOFM unable to get the stable output, which can be shown in the later experiments. The essential reason lies in the shortest Euclidean distance between the weight of cell and the train pattern as the cells compete on the output layer of the SOFM net. The Euclidean distance can not delegate the correct "distance". For example, for a frequency –agile radar, the large variety range cause long distance between two patterns of the same type radar. So the two patterns will be responded to different cells wrongly. A new definition of distance is proposed in this paper, which "debases" the effect of the problem of not being able to converge and slow convergence speed which are caused by one or more large variety-range parameters.

The definition of distance is as follows.

The distance between the k pattern and the j cell is defined as:

$$d'(x^k, m^j) = \begin{cases} \sum_{n=1}^{N} \alpha_n \left| \frac{x_n{}^k}{\max\{x_n\}} - m_n{}^j \right| & \max\{x_n\} \neq 0 \\ 0 & \max\{x_n\} = 0 \end{cases} \tag{2}$$

Where:

n : the no. n parameter in one pattern;

x^k : k input pattern, k=1,..,K;
m^j : the weight of the j cell on the output layer; j=0,1,2, …, M;
α_n: the proportion of n parameter, which is get by many valid experiments;
x_n^k: the n parameter in the x^k pattern;
m_n^j : the weight between the n parameter of the input pattern and the j cell of the output layer;
$\max(x_n)$: the maximum value among the n parameter in all input patterns.

Such "distance" defined here cant be understood by normal distance sense. It means the some measure between the input pattern and the given cell in the output layer, which is the base to modify the weights of each cell. We find such "distance" proposed here is very useful for the following experiments, although we cant prove it by strict math method for a while.

3.2 Input Pattern and Improved Algorithm

The input pattern of SOFM net is P*N matrix, signed as X^p here p is the serial number of patterns, and P represents the total number of patterns, N as the length of each pattern. There are M cells in the output layer, M>>P;

The "distance" defined by this paper (see equation (2))is used here to show the matching degree of the input pattern vector x and m_j (j=1,2,…,M).

If one cell of the output matches the input vector x, signed as **c**, as follows:

$$d'(x-m_c)=\min_j d'(x-m_j) \quad j=1,...,M \tag{3}$$

The output of **c** is:

$$y_c=\max_j y_j \quad j=1,2,...M \tag{4}$$

The improved algorithm is as follows:

Step 1: Initialize the weight as small random value, $m_j = m_j(0)$;
Step 2: Set one pattern at random among patterns $x^1, x^2,..., x^p$ as the input of Kohonen SOFM.
Step 3: Calculate the "distance" between x^r and the weight m_j according to equation (2), get the smallest distance $\|d'\| = \|x^k - m_j\| \quad j=1,2,...,M$, according to the y_c output as the "win" cell;
Step 4: Revise the weight m_j as follows:

$$\begin{cases} m_j(t+1)=m_j(t)+a_0(t)(x-m_j(t)) & j\in N_c(t) \\ m_j(t+1)=m_j(t) & j\notin N_c(t) \end{cases} \tag{5}$$

Where

$$a_0(t) = A_0 \exp(-t / \tau) \tag{6}$$

A_0 , τ are constant.

Where $N_c(t)$ is the revised zone, and it is larger at the beginning centered by y_c .Then it will decrease as follows:

$$N_c(t) = A_1 + A_2 e^{-t/\tau_1} \tag{7}$$

A_1 , A_2 and τ_1 are constant.

Step 5: Back to Step 2, till the output of the "excited" cell in the output layer becomes stable or the maxim epochs reach.

4 Computer Simulation

In ESM environments, the normal parameters used for sorting are angle of arrival (AOA), radio frequency (RF), pulse width (PW), and pulse repetition interval (PRI). For AOA, it is the only one parameter which is not affected by the radar signal itself, so it can be one of the parameters for sorting; For RF and PW, they are all stable enough on some condition, so it can also be chosen to be one parameter for sorting; For PRI, it is the important parameter to identify, so it can be one for sorting. Four representative types of radar are chosen for experiments which are listed in table 1.

Table 1. The four types of radar for experiments

Radar Type	AOA (°)	RF (MHz)	PW (us)	Radar System	PRI (us)
1	32	3030.2	1.52	Stable	315
2	43	3100.3	1.86	Frequency-agile	145
3	112	3600.5	2.60	PRI stagger	250
4	86	2680.8	1.06	Frequency-jump	125

Firstly each type generates 10, 20, 30, 40, 50 patterns to compose the input radar signals. For AOA, they follow as follows: N(32,3), N(43,3), N(112,3), N(86,3); For RF, they follow as follows: N(3030.2,3), N(3100.3, 100), N(3600,5,3), N(2680.8,200); For PRI, they follow as follows: N(115,2), N(145,2), N(150,15), N(125,3).

The α =[0.95,0.76,0.94,0.86],which is got by experiments in some ESM environments. The Euclidean distance and the distance defined in this paper are both used in the simulation. The total cell number in the output layer is 6 cells in line arrangement.

The clustering rate is the in-class distance in the patterns generated by one original radar type. If all the patterns generated by one original radar type cluster to one class, the total clustering rate will be 100%. The partition rate is the out-class distance between each class which clustering. If different patterns generated by different original radar type cause different "win" cells in the output layer, the partition rate will be 100%.

Table 2 shows that with the pattern number generated by each radar type, the clustering rate and partition rate will reach some level.

Table 2. Simulation Result of Differnt Definitions of Distance after 500 epochs

Pattern Nums \ Results	Euclidean Distance		Defined Distance	
	Clustering Rate	Partition Rate	Clustering Rate	Partition Rate
10	32.1%	11.3%	95.0%	94.0%
20	62.0%	30.2%	98.2%	93.7%
30	65.2%	32.0%	97.5%	92.3%
40	49.5%	42.3%	96.7%	90.8%
50	80.0%	45.1%	99.4%	87.9%

The result shows the improvement of clustering ability, compared with traditional distance definition. And with the increasing of pattern number generated by the original radar parameters, the clustering rate and recognition rate will increase. It also can show the Partition Rate decrease with the increasing of pattern number, which is caused by the "distance" defined. In ESM system, 15-25 patterns is used commonly.

Secondly we chose the case which 20 patterns generated by one original radar type. And after different average epochs, we check the recognition rate. It shows by figure 2 as follows.

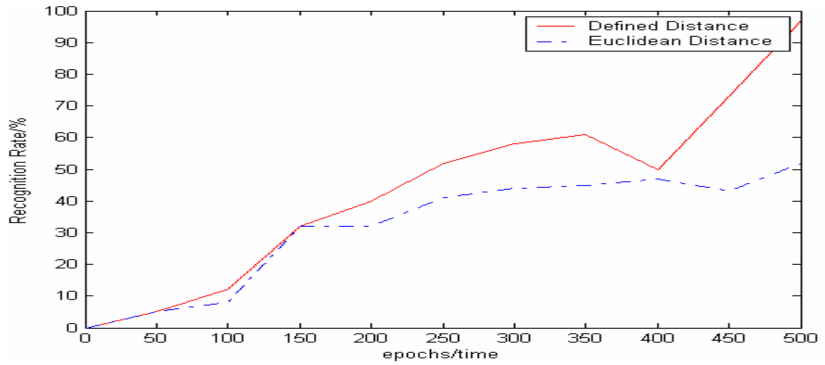


Fig. 2. the learning curve, red line shows the defined "distance" recognition rate, and the dot blue line shows the traditional one. After 500 epochs averagely, the red one will be above 95%.

During the experiences, we tried all the possible parameters of radar, the method used by this paper holds true. So the confidence intervals cover all the possible value of parameters of radar.

5 Conclusion

This paper proposes one revised SOFM net to solve the problem met when it is used in the field of radar pulse sorting in ESM system. The Euclidean distance definition is revised to fit the signal character of ESM environments. Finally, the computer simulation results show the revised network can reach the expected target. Although short of strict math prove of such “distance”, but it is very useful for such special ESM system.

References

[1] CHAN, Y. T., CHAN, F., HASSAN, H. E., Performance Evaluation of ESM Deinterleaving using TOA Analysis, The 14th International Conference on Microwave, Radar and Wireless Communications, Vol. 2, PP. 341-350, Gdansk, Poland, May, 2002.

[2] P. S. RAY, A Novel Pulse TOA Analysis Technique for Radar Identification, IEEE TRANSACTION ON AEROSPACE AND ELECTRNIC SYSTEMS VOL. 34, NO. 3 PP. 716-721, JULY 1998.

[3] Kohonen, T. (1984-1988) Self-Organization and associative memory, New York: Springer –Verla

[4] Nasrabadi, N. M., & Feng, Y.(1988) Vector quantization of images based upon the Kohonen self-organizing feature maps. In IEEE Inter. Conf. on Neural Networks (pp 1101-1108), San Diego

[5] Tsu-chang lee, Allen M.P. (1990) Adaptive Vector Quantization Using a Self-Development, Neural Network IEEE Journal on Selected Areas in Communication Vol.8 PP 1458-1471

[6] D.S. Bradrun, (1989) Reducing transmission error effects using a self-organizing network, Int. Joint. Conf. on Neural Networks, Washington, DC, June 18 22, 1989

[7] S. Carrato, “Image vector quantization using ordered codebooks: properties and application”, Signal Processing, 40(1):87-103,1994

[8] H. Ritter and K. Schulten, “Convergence properties of Kohonen’s topology conserving maps: Fluctuation, stability, and dimension selection”, Biological Cybernetics, 60:59-71,1988

[9] Specht D F. Generation of Polynomial Discriminant Function for Pattern Recognition. IEEE Trans on Electric Computers, 1967, EC-16:308

Unscented Kalman Filter-Trained MRAN Equalizer for Nonlinear Channels

Ye Zhang, Jianhua Wu*, Guojin Wan, and Yiqiang Wu

Electronic and information school of Nanchang University, Nanchang, China
zhye901@126.com, jhwu@ncu.edu.cn

Abstract. In this paper, the application of minimal resource allocation network (MRAN) trained with Unscented Kalman Filter (UKF) to the nonlinear channel equalization problems was discussed. Using novel criterion and prune strategy, the algorithm uses online learning, and has the ability to grow and prune the hidden neurons to realize a minimal network structure. Simulation results show that the equalizer is well suited for nonlinear channel equalization problems and the proposed equalizer required short training data to attain good performance.

1 Introduction

In the digital communication system, intersymbol interference (ISI) is a limiting factor in several communication environments. To achieve reliable communication in these situations, channel equalization is necessary to eliminate ISI. Fig.1 depicts the typical digital baseband transmission system; the channel model takes into account the effects of the transmitter, the transmission medium, and the receiver. The transmitted symbol $s(n)$ is assumed to be an equiprobable and independent binary sequence taking values either 1 or -1. The channel output $x(n)$ is corrupted by additive zero mean Gaussian noise $v(n)$. Here n is the time index. The nonlinear channel in a digital communication system, shown in Fig.1, can be described by:

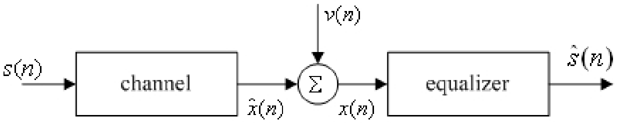


Fig. 1. Schematic of data transmission system

$$\begin{aligned} x(n) &= r(n) + k_1 r^2(n) + k_2 r^3(n) + k_3 r^4(n) + v(n) \\ r(n) &= \sum_{k=0}^{L} h(k)s(n-k) \end{aligned} \tag{1}$$

[*] Corresponding author.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 560 – 567, 2006.

where k_1, k_2,, k_3 are constants. The linear component, $H(z)=\sum_{k=0}^{L} h(k)z^{-k}$, of the channel can be modeled as a finite impulse response filter, where L is the order of the channel impulse response. The higher-order components of the linear channel are added to it to produce the nonlinear effect. In the absence of noise, the channel output takes only finite number of possible values. There are $N_s = 2^{L+m}$ possible combinations or channel states for the vector $\hat{\mathbf{x}}(n)=[\hat{x}(n),\cdots \hat{x}(n-m+1)]^T$. These output vectors are also referred to as desired channel states, and are partitioned into different classes, $\mathbf{X}^{+}_{m,d}$ and $\mathbf{X}^{-}_{m,d}$, for $s(n-d)=1$ or $s(n-d)=-1$ respectively. Here d is time delay. In the presence of noise, the channel outputs will form clusters around each of these desired channel states, and the noisy observation vector $\mathbf{x}(n)=\begin{bmatrix}x(n) & \cdots & x(n-m+1)\end{bmatrix}^T$ is used to estimate the input signal $s(n-d)$, according to the Bayesian theory. The equalization may be considered as a pattern classification problem. The associated Bayesian risk function is

$$f(\mathbf{x}(n))=\sum_{i\in N_s^+}\exp(-\left\|\mathbf{x}(n)-\mathbf{c}_i^+\right\|^2/2\sigma^2)-\sum_{j\in N_s^-}\exp(-\left\|\mathbf{x}(n)-\mathbf{c}_j^-\right\|^2/2\sigma^2) \tag{2}$$

where N_s^+ and N_s^- are the number of $\mathbf{c}_i^+$ and $\mathbf{c}_j^-$ states in $\mathbf{X}^{+}_{m,d}$ and $\mathbf{X}^{-}_{m,d}$, respectively, and σ^2 is the noise variance. The optimal decision boundary is defined by $f(\mathbf{x}(n))=0$.

Because that neural network is well suited for solving nonlinear classification problems, multilayer feedforward neural networks, radial basis function (RBF) networks and recurrent neural networks have gained popularity in their use for equalization problems. In [4], minimal resource allocation network (MRAN) was used for channel equalization. The MRAN has the same structure as a RBF network and has the ability to grow and prune the hidden neurons to achieve a compact network. Several training algorithms have been used to train RBF network, including gradient descent, back propagation (BP) [1], extended Kalman filter (EKF) [4], and so on. Major disadvantage of gradient descent and BP methods are slow convergence rates and the long training symbols required [1]. The EKF can be used to determine the centers, radius and weights of the RBF network; the advantage of this method is not necessary to estimate the channel order. But the EKF algorithm provides first-order approximations to optimal nonlinear estimation through the linearization of the nonlinear system. These approximations can include large errors in the true posterior mean and covariance of the transformed (Gaussian) random variable, which may lead to suboptimal performance and sometimes divergence. The unscented Kalman filter (UKF) is an alternative to the EKF algorithm and provides third-order approximation of process and measurement errors for Gaussian distributions and at least second-order approximation for non-Gaussian distributions [2]. Consequently, the UKF may have better performance than the EKF. In this paper, we use UKF to estimate the parameters of the MRAN network. In our simulation, the performance of the MRAN equalizer trained with UKF is superior to the MRAN equalizer trained with EKF.

2 MRAN Channel Equalizer and EKF

The MARN is a sequential learning RBF network and the MRAN algorithm uses online learning, and has the ability to grow and prune the hidden neurons to realize a minimal network structure [4]. Fig.2 shows a schematic of a channel equalizer based on RBF network. Two layers; hidden layer consisting of N local units, and a linear output layer form the RBF neural network. The output is given by:

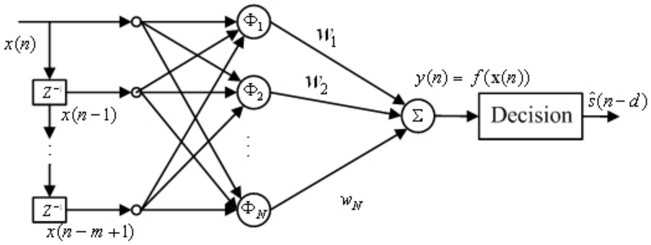


Fig. 2. Architecture of MRAN equalizer

$$y(n)=f(\mathbf{x}(n))=\sum_{i=1}^{N}w_i(n)\Phi_i(\mathbf{x}(n)) \tag{3}$$

where input vector $\mathbf{x}(n)=\left[x(n) \quad \cdots \quad x(n-m+1)\right]^T$, $\Phi_i(\cdot)$ denotes the mapping performed by a local unit, and $w_i(n)$ is the weight associated with that unit. The basis function is usually selected as Gaussian function

$$\Phi_i=\exp(-\left\|\mathbf{x}(n)-\mathbf{c}_i(n)\right\|^2 / \sigma_i^2(n)) \tag{4}$$

where $\mathbf{c}_i(n)$ and $\sigma_i(n)$ will be referred to as the center and radius, respectively. Comparing the network response (3) with the optimal Bayesian equalizer filter (2), it is obvious that they have the same structure. The RBF network is therefore an ideal processing means to implement the optimal Bayesian equalizer [4]. It can be seen that the design of a RBF requires several decisions, including the centers $\mathbf{c}_i(n)$, the radius $\sigma_i(n)$, the number *N*, and weight $w_i(n)$. In MRAN algorithms, the number *N* of neurons in the hidden layer does not estimate, so the order of the channel is not necessary to be estimated, the network is built based on certain growth criteria. Other network parameters, such as $\mathbf{c}_i(n),\sigma_i(n),w_i(n)$, can be adapted using the EKF [4].

The MRAN network begins with no hidden neuron. As input vector $\mathbf{x}(n)$ are sequentially received, the network builds up based on certain growth and pruning criteria [6]. The following three criteria decide whether a new hidden neuron should be added to the network:

$$\begin{aligned}&\left\|\mathbf{x}(n)-\mathbf{c}_j(n)\right\|>\varepsilon(n)\\&e(n)=s(n)-f(\mathbf{x}(n))>e_{\min}\\&e_{rms}(n)=\sqrt{\frac{\sum_{i=n-M+1}^{n}\left[d(n)-f(\mathbf{x}(n))\right]^2}{M}}>e'_{\min}\end{aligned}\tag{5}$$

where $\mathbf{c}_j(n)$ is a centre of the hidden neuron that is nearest to $\mathbf{x}(n)$, the data that was just received. $\varepsilon(n), e_{\min}$ and $e'_{\min}$ are threshold to be selected appropriately. M represents the size of a sliding data window that the network has not met the required sum squared error specification. Only when all these criteria are met a new hidden node added to the network. The parameters associated with it:

$$w_{N+1}=e(n),\quad \mathbf{c}_{N+1}=\mathbf{x}(n),\quad \sigma_{N+1}=\kappa\left\|\mathbf{x}(n)-\mathbf{c}_j(n)\right\|\tag{6}$$

where κ is an overlap factor that determine the overlap of the response of the hidden neuron in the input space. When an input to the network does not meet the criteria for adding a new hidden neuron, EKF will be used to adjust the parameters $\boldsymbol{\theta}=\left[w_1,\mathbf{c}_1^T,\sigma_1,\cdots,w_N,\mathbf{c}_N^T,\sigma_N\right]^T$ of the network. The network model to which the EKF can be applied is

$$\begin{aligned}\boldsymbol{\theta}(n+1)&=\boldsymbol{\theta}(n)+\boldsymbol{\omega}(n)\\y(n)=f(\mathbf{x}(n))&=\sum_{i=1}^{N}w_i(n)\Phi_i(\mathbf{x}(n))+v(n)\\&=g(\boldsymbol{\theta}(n),\mathbf{x}(n))+v(n)\end{aligned}\tag{7}$$

where $\boldsymbol{\omega}(n)$ and $v(n)$ are artificial added noise processes, $\boldsymbol{\omega}(n)$ is the process noise, $v(n)$ is the observation noise. The desired estimate $\hat{\boldsymbol{\theta}}(n)$ can be obtained by the recursion

$$\begin{aligned}&\hat{\boldsymbol{\theta}}(n)=\hat{\boldsymbol{\theta}}(n-1)+\mathbf{K}(n)e(n)\\&\mathbf{K}(n)=\mathbf{P}(n-1)\mathbf{a}(n)\left[\mathbf{R}(n)+\mathbf{a}^T(n)\mathbf{P}(n-1)\mathbf{a}(n)\right]^{-1}\\&\mathbf{P}(n)=\left[\mathbf{I}-\mathbf{k}(n)\mathbf{a}^T(n)\right]\mathbf{P}(n-1)+\mathbf{Q}(n)\mathbf{I}\end{aligned}\tag{8}$$

where $\mathbf{K}(n)$ is the Kalman gain, $\mathbf{a}(n)$ is the gradient vector and has the following form

$$\mathbf{a}^T(n)=\left.\frac{\partial g(\boldsymbol{\theta},\mathbf{x}(n))}{\partial\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}(n)}\tag{9}$$

$\mathbf{P}(n)$ is the error covariance matrix, $\mathbf{R}(n)$ and $\mathbf{Q}(n)$ are the covariance matrices of the artificial noise processes $\boldsymbol{\omega}(n)$ and $v(n)$, respectively. When a new hidden neuron is added the dimensionality of $\mathbf{P}(n)$ is increased by

$$\mathbf{P}(n)=\begin{pmatrix}\mathbf{P}(n-1) & 0\\ 0 & \mathbf{P}_0\mathbf{I}\end{pmatrix}\tag{10}$$

The new rows and columns are initialized by $\mathbf{P}_0$. $\mathbf{P}_0$ is an estimate of the uncertainty in the initial values assigned to the parameters. The dimension of identity matrix **I** is equal to the number of new parameters introduced by adding a new hidden neuron.

In order to keep the MRAN in a minimal size and a pruning strategy is employed [4]. According to this, for every observation, each normalized hidden neuron output value $r_k(n)$ is examined to decide whether or not it should be removed.

$$\begin{aligned} &o_k(n) = w_k(n)\exp(-\|\mathbf{x}(n)-\mathbf{c}_k(n)\|^2/\sigma_k^2(n)) \\ &r_k(n) = \left\| \frac{o_k(n)}{o_{\max}(n)} \right\|, \quad k = 1,\cdots,N \end{aligned} \tag{11}$$

where $o_k(n)$ is the output for *k*th hidden neuron at time *n* and $o_{\max}(n)$, the largest absolute hidden neuron output value at *n*. These normalized values are compared with a threshold δ and if any of them falls below this threshold for *M* consecutive observation then this particular hidden neuron is removed from the network.

3 Using UKF for Training the MRAN Channel Equalizer

The EKF described in the previous section provides first-order approximations to optimal nonlinear estimation through the linearization of the nonlinear system. These approximations can include large errors in the true posterior mean and covariance of the transformed (Gaussian) random variable, which may lead to suboptimal performance and sometimes divergence [1]. The unscented Kalman filter is an alternative to the EKF algorithm. The UKF provides third-order approximation of process and measurement errors for Gaussian distributions and at least second-order approximation for non-Gaussian distributions [2]. Consequently, The UKF may have better performance than the EKF. In this section, we propose the UKF algorithm to adjust the parameters of the network, when an input to the network does not meet the criteria for adding a new hidden neuron.

Foundation to the UKF is the unscented transformation (UT). The UT is a method for calculating the statistic of a random variable that undergoes a nonlinear transformation [7]. Consider propagating a random variable **x** (dimension *m*) through a nonlinear function, $\mathbf{y} = g(\mathbf{x})$. To calculate the statistic of y, a matrix χ of 2*m*+1 sigma vectors χ_i is formed as the followings:

$$\begin{aligned} &\chi_0 = \bar{\mathbf{x}} \\ &\chi_i = \bar{\mathbf{x}} + \left(\sqrt{(m+\lambda)\mathbf{P}_{xx}}\right)_i \quad i = 1,\cdots,m \\ &\chi_i = \bar{\mathbf{x}} - \left(\sqrt{(m+\lambda)\mathbf{P}_{xx}}\right)_{i-L} \quad i = m+1,\cdots,2m \\ &W_0^m = \lambda/(m+\lambda) \\ &W_0^c = \lambda/(m+\lambda) + (1-a^2+\beta) \\ &W_i^m = W_i^c = 1/(2m+2\lambda) \qquad i = 1,\cdots,2m \end{aligned} \tag{12}$$

where $\bar{\mathbf{x}}$ and $\mathbf{P}_{xx}$ are the mean and covariance of $\mathbf{x}$, respectively, and $\lambda = a^2(m+\rho)-m$ is a scaling factor. a determines the spread of the sigma points around $\bar{\mathbf{x}}$ and usually set to a small positive value, typically in the range $0.001 < a < 1$. ρ is a secondary scaling parameter which is usually set to 0, and β is used to take account for prior knowledge on the distribution of **x,** and $\beta = 2$ is the optimal choice for Gaussian distribution[8]. These sigma vectors are propagated through the nonlinear function

$$y_i = g(\chi_i) \quad i = 0,\cdots,2m \tag{13}$$

This propagation produces a corresponding vector set that can be used to estimate the mean and covariance matrix of the nonlinear transformed vector $\mathbf{y}$.

$$\begin{aligned} \bar{\mathbf{y}} &\approx \sum_{i=0}^{2m} W_i^m y_i \\ \mathbf{P}_{yy} &\approx \sum_{i=0}^{2m} W_i^c \left(y_i - \bar{\mathbf{y}}\right)\left(y_i - \bar{\mathbf{y}}\right)^T \end{aligned} \tag{14}$$

From the state-space model of the MRAN given in (7), when an input to the network does not meet the criteria for adding a new hidden neuron, we can use the UKF algorithm to adjust the parameters of the network. The algorithms are summarized below.

Initialized with:

$$\begin{aligned} \hat{\boldsymbol{\theta}}(0) &= E[\boldsymbol{\theta}] \\ \mathbf{P}(0) &= E\left[(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(0)(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(0)^T\right] \end{aligned} \tag{15}$$

The sigma-point calculation:

$$\begin{aligned} &\boldsymbol{\Gamma}(n) = (m+\lambda)(\mathbf{P}(n) + \mathbf{Q}(n)) \\ &\mathbf{W}(n) = \left[\hat{\boldsymbol{\theta}}(n), \hat{\boldsymbol{\theta}}(n) + \sqrt{\boldsymbol{\Gamma}(n)}, \hat{\boldsymbol{\theta}}(n) - \sqrt{\boldsymbol{\Gamma}(n)}\right] \\ &D(n) = g(\mathbf{W}(n), \mathbf{x}(n)) \\ &y(n) = g(\hat{\boldsymbol{\theta}}(n), \mathbf{x}(n)) \end{aligned} \tag{16}$$

Measurement update equations:

$$\begin{aligned} \mathbf{P}_{\mathbf{yy}}(n) &= \sum_{i=0}^{2m} W_i^c (D_i(n) - \bar{\mathbf{y}}(n))(D_i(n) - \bar{\mathbf{y}}(n))^T + \mathbf{R}(n) \\ \mathbf{P}_{\boldsymbol{\theta}\mathbf{y}}(n) &= \sum_{i=0}^{2m} W_i^c (W_i(n) - \hat{\boldsymbol{\theta}}(n))(W_i(n) - \hat{\boldsymbol{\theta}}(n))^T \end{aligned} \tag{17}$$

$$\mathbf{K}(n) = \mathbf{P}_{\boldsymbol{\theta}\mathbf{y}}(n)\mathbf{P}_{\mathbf{yy}}^{-1}(n) \tag{18}$$

$$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) + \mathbf{K}(n)e(n) \tag{19}$$

$$\mathbf{P}(n+1) = \mathbf{P}(n) - \mathbf{K}(n)\mathbf{P}_{yy}(n)\mathbf{K}^T(n) \tag{20}$$

The parameter vector of the MRAN is update with the above equations.

4 Experiment Results and Conclusion

In the experiments, the thresholds $e_{\min}, e'_{\min}$, and ε, respectively, set as 0.22, 0.40, and 0.5, the thresholds were chosen largely by trial and error. The other parameters were set as M=10 and δ=0.1. To test the algorithm for non-linear channels, the following 2PAM nonlinear channel [4] was chosen:

$$\begin{aligned} x(n) &= r(n) + 0.2r^2(n) + v(n) \\ H(z) &= 0.3482 + 0.8704z^{-1} + 0.3482z^{-2} \end{aligned} \tag{21}$$

For the purpose of graphical display, the equalizer order is chosen as L =2. In the example, m=2**.** Thus, there will be 16 desired states for the channel output, (2^{L+m}=16). The decision delay was set to one (d=1). By using the MRAN algorithm with 500 data samples at 12dB SNR, we were able to obtain the classification boundary shown in figure 3. The continuous line shows the Bayesian boundary, while the boundary obtained by the UKF algorithm is shown by the dotted line. The MRAN centres created by the UKF algorithm are indicated by the '$*$', while the actual desired states are indicated by the '□'. The network has built up 17 hidden nodes and this is more than the 16 desired channel states. It can be seen that the Bayesian boundary is still well approximated, at the critical region, which is at the centre of the figure. At the bottom region in the figure, the network boundary deviates from the Bayesian boundary, but this can be seen to be less critical in the equalization task, from the BER curves shown in Fig. 4.

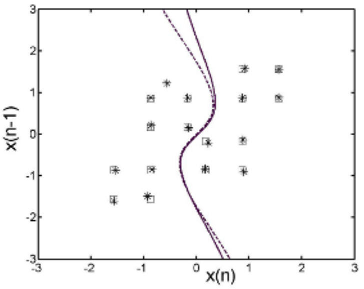


Fig. 3. Boundary and location of the equalizer

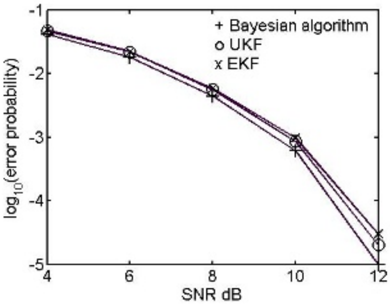


Fig. 4. The performance of the equalizers

Fig.4 shows the BER performance for the three equalizers for the channel, averaged over 20 independent trials. In each trial, the first 200 symbols are used for training and the next 10^5 symbols are used for testing. The parameter vectors of the equalizers are constant after the training stage, and then the test is continued. It is clear that the MRAN trained with UKF is better than the MRAN trained with EKF for the nonlinear channel and the performance of the MRAN trained with UKF is only slightly poorer than the Bayesian equalizer.

We have presented a MRAN equalizer trained with the UKF for nonlinear channel equalization over 2PAM signals. Simulation results show that the equalizer is well suited for nonlinear channel equalization problems. The performance of the MRAN equalizer trained with UKF has been compared with that of the ideal Bayesian equalizer, and the MRAN equalizer trained with EKF. Simulation results showed that the MRAN equalizer trained with UKF performed better than the MRAN trained with EKF. Moreover, the proposed equalizer required short training data to attain good performance.

Acknowledgment

This paper is supported by the JiangXi Province Natural Science Foundation (0511076).

References

1. Simon Haykin.: Neural networks A comprehensive foundation, Second Edition. Upper Saddle River, NJ: Prentice Hall (1999)
2. E.A. Wan and R. Van der Merwe.: The unscented Kalman filter, in Kalman filtering and neural networks. John Wiley and Sons, Inc. (2001)
3. E.A.Wan and R. Van der Merwe.; The unscented Kalman filter for nonlinear estimation. In Proc.of IEEE symposium (2000) 152-158
4. P. Chandra Kumar, P.Saratchandran and N. Sundararajan.: Minimal radial basis function neural network for nonlinear channel equalization. IEE Proc.-Vis. Image Signal Process, Vol. 147. (2000) 428-435
5. J. Lee, C. Beach, and N. Tepedelenlioglu.: Channel equalization using radial basis function network. In Proc. Of ICASSP'96, Atlanta, GA,May (1996) 797-802
6. Lu Yingwei, N Sundararajan, P Saratchandran, "Adaptive nonlinear system identification using minimal radial basis function neural networks", IEEE ICASSP*,* Vol 6, (1996) 3521-3524
7. Jongsoo Choi, C Lima, A.C., Haykin, S.; Unscented Kalman filter-trained recurrent neural equalizer for time-varying channels. ICC'03, Vol 5.(2003) 3241-3245
8. Jongsoo Choi; Lima, A.Cd.C.; Haykin, S.; Kalman filter-trained recurrent neural equalizers for time-varying channels. IEEE Transactions on Communications, Vol53. (2005)472-480

A Jumping Genes Paradigm with Fuzzy Rules for Optimizing Digital IIR Filters

Sai-Ho Yeung and Kim-Fung Man

Department of Electronic Engineering,
City University of Hong Kong, Kowloon, Hong Kong
shyeung@ee.cityu.edu.hk

Abstract. A Jumping Genes Paradigm that combines with fuzzy rules is applied for optimizing the digital IIR filters. The criteria that govern the quality of the optimization procedure are based on two basic measures. A newly formulated performance metric for the digital IIR filter is formed for checking its performance while its system order which usually reflects upon the required computational power is also adopted as another objective function for the optimization. The proposed scheme in this paper was able to obtain frequency-selective filters for lowpass, highpass, bandpass and bandstop with better performance than those previously obtained and the filter system order was also optimized with lower possible number.

Keywords: IIR Filter, Genetic Algorithm, Fuzzy Logic.

1 Introduction

In digital signal processing, the Infinite Impulse Response (IIR) Filter [1], or recursive filter is an important component for signal filtering. The traditional methods of designing the frequency-selective IIR Filters include Butterworth, Chebyshev Type 1, Chebyshev Type 2, and Elliptic function have been well reported. The improved Hierarchical Genetic Algorithm (HGA) approach for IIR filter design was presented in [2-3]. A minimum filter order that meets the specific frequency response is obtained. In this paper, a new scheme for improving the HGA approach in IIR filter design is presented. In this scheme, fuzzy rules [4] are applied in evaluating the performance of IIR filter based on the pass-band ripple and stop-band ripple. The performance metric of the filter is used as an objective function for optimizing the filter performance. A novel Evolutionary Computing Algorithm, Jumping Genes Evolutionary Algorithm (JGEA) [5-7] is adopted as the optimization scheme. The advantage of using JGEA is its ability in obtaining a set of non-dominated solutions that is close to Pareto-optimal front. Using the JGEA combining with fuzzy rules, the lowpass (LP) filter, highpass (HP) filter, bandpass filter (BP) and bandstop (BS) filter are successfully designed.

The paper is organized as follows: The fuzzy rules for evaluating the performance of IIR will be presented in Section 2. In Section 3, the JGEA scheme combining the fuzzy rules for the optimization of IIR filter will be discussed. Then, Section 4 will

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 568 – 577, 2006.

compare the proposed fuzzy scheme with other choices of objective functions for filter design. Finally, the conclusion will be given in Section 5.

2 Fuzzy Rules for Filter Performance Evaluation

In the design process of frequency-selective IIR filter, designer should specific the requirement on the tolerance of the ripple inside the pass-band and stop-band, which are denoted as δ_1 and δ_2 respectively. Fig. 1 illustrates the concepts of the tolerance scheme on the filter design for an example of a low-pass filter design. To describe the magnitude of the ripple inside the pass-band and the stop-band regions, the linguistic terms Very Small (VS), Small (S), Medium (M), Large (L), Very Large (VL) are used, where Fig. 2 shows the membership functions. Note that the membership functions are depending on the tolerance δ_1 and δ_2 for ripples inside the pass-band and the stop-band respectively.

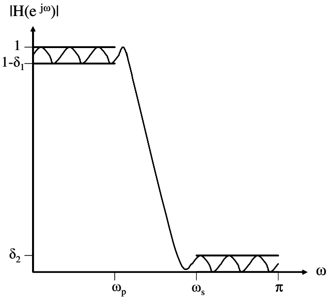


Fig. 1. Tolerance scheme for a low-pass filter design

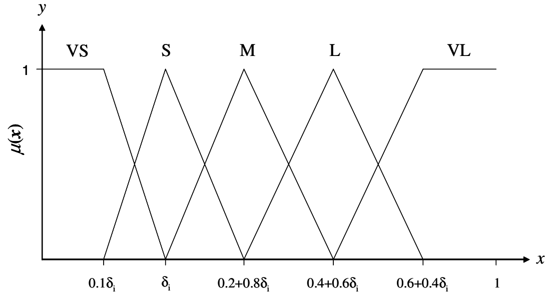


Fig. 2. Membership Function for input variable filter ripple

The linguistic terms of Very Good (VG), Good (G), Average (A), Bad (B), Very Bad (VB) are used to describe the performance of the filter. The membership function are shown in Fig. 3. The performance of the filter depends on the pass-band and stop-band ripple by fuzzy relations, where it is implemented as the fuzzy rules base in

Table 1. Note that the filter performance is above "GOOD" only if the pass-band and stop-band ripples are both "Small" or "Very Small", and the ripple is "Small" or "Very Small" when the ripple is smaller than the tolerance δ. Thus, the performance above "GOOD" indicates that both ripples in the pass-band and stop-band are smaller than the tolerance δ, and hence becomes a feasible solution for the filter design.

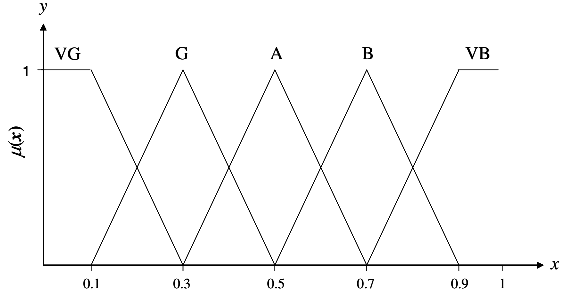


Fig. 3. Membership Function for output variable filter performance

Table 1. Fuzzy Rules Base determining the Filter Performance

Performance		Passband Ripple				
		VS	S	M	L	VL
Stopband Ripple	VS	**VG**	**G**	**M**	**B**	**VB**
	S	**G**	**G**	**M**	**B**	**VB**
	M	**M**	**M**	**M**	**B**	**VB**
	L	**B**	**B**	**B**	**B**	**VB**
	VL	**VB**	**VB**	**VB**	**VB**	**VB**

2.1 Filter Performance Metric Evaluation

The defuzzification is the process of turning the membership values on the filter performance to a single performance metric, and it is given by the following weighted-average formula:

$$Performance\ Metric = \frac{u_{VG} \times 0.1 + u_G \times 0.3 + u_A \times 0.5 + u_B \times 0.7 + u_{VB} \times 0.9}{u_{VG} + u_G + u_A + u_B + u_{VB}} \qquad (1)$$

The performance metric that has a value of 0.3 classifies that the performance is "GOOD", so a performance metric value below 0.3 indicates both ripples in the pass-band and stop-band are smaller than the tolerance, and hence it is a feasible filter solution.

The steps for obtaining a filter performance metric by fuzzy rules are summarized as below:

Step 1: From the filter coefficients, calculate the pass-band and stop-band ripples;

Step 2: Define the tolerance requirements δ_1 and δ_2 on the pass-band and the stop-band, and calculate the membership functions of the pass-band and stop-band ripples;

Step 3: Use the fuzzy rules to calculate the membership functions of the filter performance; and

Step 4: Defuzzification on the filter performance to obtain the performance metric. If the evaluated value is smaller than 0.3, the filter satisfies the tolerance requirements.

3 JGEA for IIR Filter Optimization

JGEA is applied for optimizing the IIR filters in cooperating with the fuzzy rules introduced in Section 2.

3.1 Optimization Problem Formulation

The filter performance metric obtained by the fuzzy rules are used as an objective function for optimizing the performance of the filter. Four kinds of frequency-selective filters: LP, HP, BP and BS filters are considered. The criteria that govern the performance are formulated as follows.

Minimize

$$f_1 = Performanc\ e\ Metric \tag{2}$$

$$f_2 = Filter\ Order \tag{3}$$

The objective function f_1 is the filter performance metric evaluated by the method stated in Section II, whereas f_2 is the order of the filter that determines the less required filter order of the transfer function directly. In filter design, the filter order should also be minimized so that a minimum use of computational power for filtering is ensured. This can be arranged in the form of HGA format for the chromosome structure.

3.2 Optimization Algorithm JGEA

JGEA [5-7] is a novel evolutionary algorithm for multi-objective optimization (MOEA) [8-9], where it introduces a new genetic operator using a horizontal gene transmission mechanism, i.e. jumping genes transposition. It enables the genes transfer between the individuals within the same generation. After the evaluation of the objective functions, the fast non-dominated sorting is used for ranking the solutions and crowding distance assignment [10] is used for the diversity preservation of the population pool. It has been proven to out-perform others MOEAs in various applications, such as wireless local area network in IC factory [5], radio-to-fiber repeater placement [6], and resource management in wideband CDMA systems [7].

As JGEA is applied for optimizing the IIR filters in cooperating with the fuzzy rules, the flowchart of the optimization process is shown in Fig. 4.

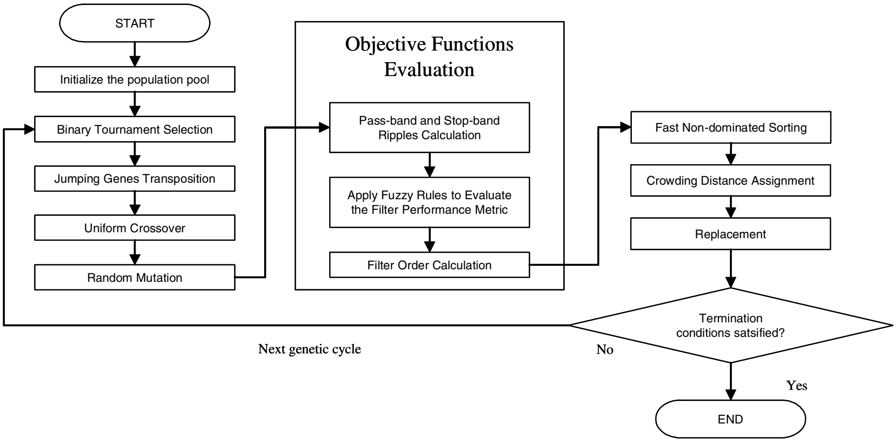

START
Initialize the population pool
Binary Tournament Selection
Jumping Genes Transposition
Uniform Crossover
Random Mutation
Objective Functions Evaluation
Pass-band and Stop-band Ripples Calculation
Apply Fuzzy Rules to Evaluate the Filter Performance Metric
Filter Order Calculation
Fast Non-dominated Sorting
Crowding Distance Assignment
Replacement
Termination conditions satsified?
Next genetic cycle
No
Yes
END

Fig. 4. Flowchart of the Optimization Process

3.3 Optimal Filter Solutions

Given that with the selected values of tolerance $\delta_1 = 0.10785$ and $\delta_2 = 0.17783$ for the filter design requirement, the JGEA optimization scheme produced the filter solutions with different filter order, of which all the non-dominated solutions are shown in Fig. 5. The solutions with filter performance value lower than 0.3 in the figure are considered the feasible solutions, as this indicates that the ripples is inside both the pass-band and the stop-band satisfy the tolerance requirement. The higher order filter may be preferred than the lower order filters for achieving better performance, but low order solution has the advantage of lower computational power requirement.

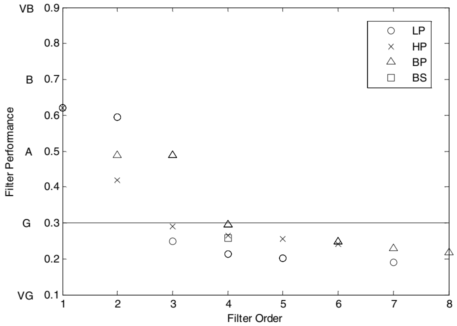


Fig. 5. Non-dominated solution fronts

The obtained LP, HP, BP and BS filters that meet the design requirements are shown in Fig. 6, 7, 8 and 9 respectively. Whereas the typical transfer functions of the optimized lowest order filters are listed below:

$$H_{LP}(z) = \frac{0.155815z^3 - 0.129247z^2 + 0.088092z + 0.049638}{z^3 - 1.993688z^2 + 1.645483z - 0.487495} \tag{4}$$

$$H_{HP}(z) = \frac{0.070203z^3 - 0.024019z^2 + 0.000609z - 0.046465}{z^3 + 1.956504z^2 + 1.550864z + 0.440415} \tag{5}$$

$$H_{BP}(z) = \frac{0.14975z^4 - 0.031995z^3 - 0.15194z^2 + 0.028066z + 0.023288}{z^4 - 0.091133z^3 + 1.1018z^2 - 0.052952z + 0.46562} \tag{6}$$

$$H_{BS}(z) = \frac{0.35657z^4 - 0.002655z^3 + 0.70392z^2 - 0.0026545z + 0.35644}{z^4 + 0.000732z^3 + 0.23193z^2 + 0.008751z + 0.27331} \tag{7}$$

For comparison purpose, the results obtained originally from HGA filters design methodology in [2-3] are made to compare with the solutions produced by JGEA. The magnitude of the pass-band ripple and the stop-band ripple are marked in the Fig. 6, 7, 8 and 9 for ease of reference.

Table 2. Filters Design Criteria

Filter Type	Pass-band	Stop-band	Iteration for EA	Filter Order Search Range
LP	$0 \le \lvert\omega\rvert \le 0.2\pi$	$0.3\pi \le \lvert\omega\rvert \le \pi$	500	[1, 15]
HP	$0 \le \lvert\omega\rvert \le 0.7\pi$	$0.8\pi \le \lvert\omega\rvert \le \pi$	500	[1, 15]
BP	$0.4\pi \le \lvert\omega\rvert \le 0.6\pi$	$0 \le \lvert\omega\rvert \le 0.25\pi$ $0.75\pi \le \lvert\omega\rvert \le \pi$	5000	[2, 15]
BS	$0 \le \lvert\omega\rvert \le 0.25\pi$ $0.75\pi \le \lvert\omega\rvert \le \pi$	$0.4\pi \le \lvert\omega\rvert \le 0.6\pi$	5000	[4, 15]

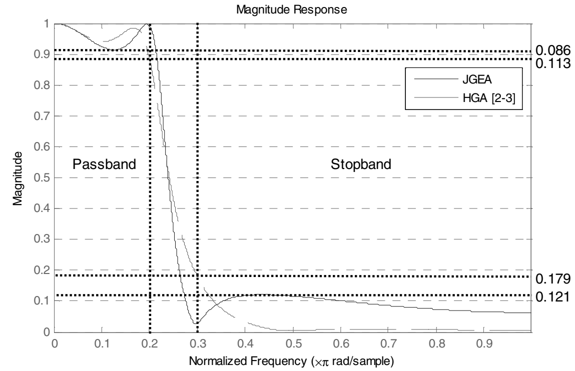


Fig. 6. Optimized Lowpass Filter

The comparative results are tabulated in Table 3. It is clear that the LP, BP and BS filters designed by JGEA with fuzzy rules were found to have smaller ripples than those designed by HGA. Furthermore, JGEA was able to obtain a lower order with smaller ripples for BP filter. Thus, the new proposed method, applying JGEA and fuzzy rules, can be considered as a better alternative method for IIR filter design.

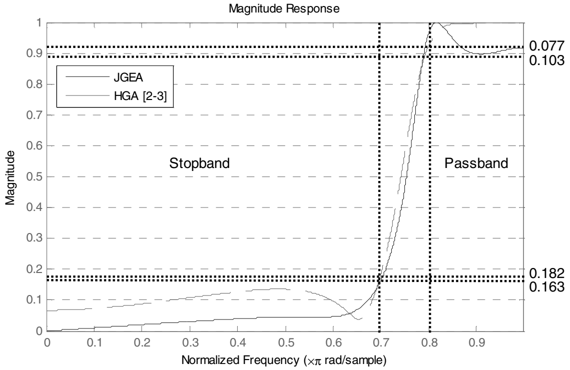


Fig. 7. Optimized Highpass Filter

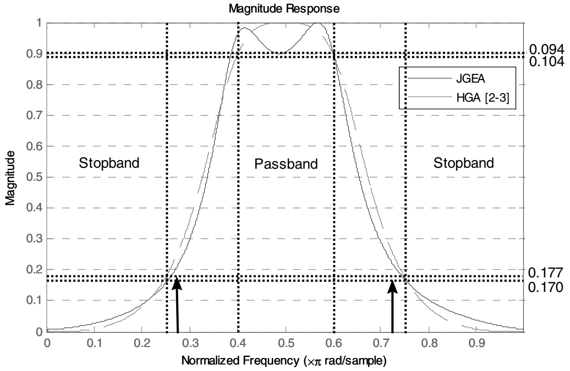


Fig. 8. Optimized Bandpass Filter

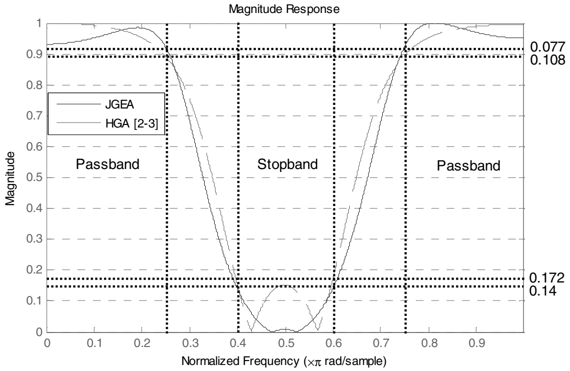


Fig. 9. Optimized Bandstop Filter

Table 3. Optimized Filters Performance

Filter Type	Design Methodology					
	JGEA with Fuzzy Rules			HGA [2-3]		
	Pass-band Ripple	Stop-band Ripple	Filter Order	Pass-band Ripple	Stop-band Ripple	Filter Order
LP	0.086	0.121	3	0.113	0.179	3
HP	0.103	0.163	3	0.077	0.182	3
BP	0.094	0.17	4	0.104	0.177	6
BS	0.077	0.14	4	0.108	0.172	4

4 Discussion on Other Choices of Objectives Functions

In this section, other possible choices of objective functions for IIR filter design will be discussed. Two optimization schemes which use three objective functions will be investigated. They will be compared with the proposed scheme where fuzzy evaluated objective function is used.

4.1 Three Objectives Optimization Scheme

Now, consider another optimization scheme where pass-band and stop-band ripples are minimized separately instead of using a single fuzzy filter quality measure. Hence, in total three optimization objectives are used: f_1 and f_2 minimize the maximum magnitude of pass-band ripple and stop-band ripple, respectively, whereas f_3 minimize the filter order.

Fig. 10 shows all the non-dominated solutions for low-pass filter design where 2000 generations of evolution is set as the termination criteria. However, only one solution satisfies the tolerance requirement, where the pass-band and stop-band ripples are smaller than $\delta_1 = 0.10785$ and $\delta_2 = 0.17783$, respectively. It should be noted that solutions which do not satisfy the tolerance requirement can be also classified as non-dominated solutions throughout the optimization process. For instance, solutions in Fig. 10 do not dominate each other, but only one of them is feasible solution. Hence, this optimization scheme is not effective as compared with the proposed fuzzy scheme.

4.2 Original Proposed HGA Filter Design Scheme in [2-3]

The original proposed HGA filter optimization scheme in [2-3] also use three objective functions, but with some difference: f_1 and f_2 minimize the summation of excessive filter ripple at different frequency points, in pass-band and stop-band, respectively, whereas f_3 minimize the filter order. This scheme has successfully designed the filter satisfying the tolerance requirement with minimum filter order. However, whenever the pass-band and stop-band satisfy the user defined tolerance

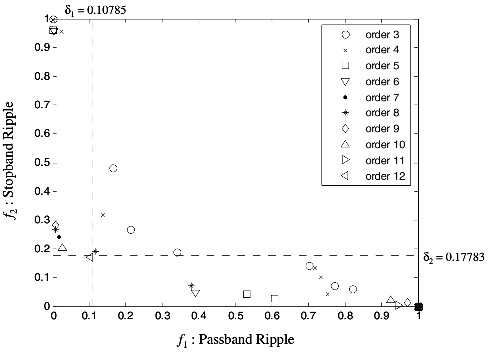


Fig. 10. Non-dominated solutions obtained by the 3 objectives optimization scheme

requirement, the objective values will be $f_1 = 0$ and $f_2 = 0$. In this case, the magnitude of the ripples inside the pass-band and stop-band can not be given by the objective values, but only it is given that the filter satisfies the tolerance requirement. As a result, this optimization scheme will not further minimize the ripples after the filter has satisfied the tolerance requirement. As a comparison, the proposed fuzzy evaluated objective function can further minimize the ripples when the filter has already satisfied the tolerance requirement: when the filter ripple exactly equals the tolerance requirement, $f_1 = 0.3$; when the filter ripple is smaller than the tolerance requirement, $f_1 < 0.3$.

5 Conclusion

In this paper, the use of JGEA with fuzzy rules for the optimization of IIR filters has been demonstrated. Given with the tolerance requirements, the designed LP, BP and BS and BP filters were all found to have smaller ripple than that originally designed filter by HGA approach while a newly discovered lower order for BP filter by JGEA was obtained. Moreover, the obtained Pareto-optimal solutions as indicated in Fig. 5 also provided useful tradeoff information between the filter performance and the filter order in which this allows the designer to choose an appropriate solution to meet the design requirements. Also, it is demonstrated that the proposed fuzzy optimization scheme is better than some other schemes which uses three objectives functions.

References

1. Bellanger, M.: Digital processing of signals : theory and practice. 3rd edn. John Wiley & Sons Ltd. Chichester, England (2000)
2. Tang, K. S., Man, K. F., Kwong S., Liu, Z. F.: Design and optimization of IIR filter structure using hierarchical genetic algorithms. IEEE Trans on Industrial Electronics, Vol. 45, Issue 3 (1998) 481-487
3. Man, K. F., Tang, K. S., Kwong S.: Genetic Algorithms: Concepts and Design. Springer-Verlag, Berlin Heidelberg London (1999)

4. Chen, G. R., Pham, T. T.: Introduction to Fuzzy Systems. Taylor & Francis Group, LLC (2006)
5. Chan, T. M., Man, K. F., Tang, K. S., Kwong, S.: Optimization of wireless local area network in IC factory using a jumping-gene paradigm. In: Proceedings of the 3rd International IEEE Conference on Industrial Informatics (INDIN2005), Perth, Western Australia (2005) 773-778
6. Chan, T. M., Man, K. F., Tang, K. S., Kwong, S.: Multiobjective optimization of radio-to-fiber repeater placement using a jumping gene algorithm. In: Proceedings of the IEEE International Conference on Industrial Technology (ICIT2005), Hong Kong, China (2005) 291-296
7. Chan, T. M., Man, K. F., Tang, K. S., Kwong, S.: A Jumping gene algorithm for multiobjective resource management in wideband CDMA systems. The Computer Journal, Vol. 48, No. 6 (2005) 749-768
8. Coello, C. A., Van Veldhuizen D. A., Lamont G.B.: Evolutionary Algorithms for Solving Multiobjective Problems. Kluwer Academic Publishers (2002)
9. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons Ltd. Chichester, England (2001)
10. Deb, K., Pratap, A., Agrawal, S., and Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2 (2002) 182-197

Practical Denoising of MEG Data Using Wavelet Transform

Abhisek Ukil

Tshwane University of Technology, Pretoria, 0001, South Africa
abhiukil@yahoo.com

Abstract. Magnetoencephalography (MEG) is an important noninvasive, non-hazardous technology for functional brain mapping, measuring the magnetic fields due to the intracellular neuronal current flow in the brain. However, the inherent level of noise in the data collection process is large enough to obscure the signal(s) of interest most often. In this paper, a practical denoising technique based on the wavelet transform and the multiresolution signal decomposition technique is presented. The proposed technique is substantiated by the application results using three different mother wavelets on the recorded MEG signal.

1 Introduction

Magnetoencephalography (MEG) is completely noninvasive, non-hazardous technology for functional brain mapping. Every current generates a magnetic field, and following this same principle in the nervous system, the longitudinal neuronal current flow generates an associated magnetic field. MEG measures the intercellular currents of the neurons in the brain giving a direct information on the brain activity, spontaneously or to a given stimulus. That is, MEG detects weak extracranial magnetic fields in the brain, and allows determination of their intracranial sources [1].

Unlike Computed Tomography (CT) or Magnetic Resonance Imaging (MRI), which provide structural/anatomical information, MEG provides functional mapping information. By measuring these magnetic fields, scientists can accurately pinpoint the location of the cells that produce each field. In this way, they can identify zones of the brain that are producing abnormal signals. These spatiotemporal signals are used to study human cognition and, in clinical settings, for preoperative functional brain mapping, epilepsy diagnosis and the like.

One common method of collecting functional data involves the presentation of a stimulus to a subject. However, most often the inherent noise level in the data collection process is large enough to obscure the signal(s) of interest. In order to reduce the level of noise the stimulus is repeated for as many as 100-500 trials, the trials are temporally aligned based on the timing of the stimulus presentation, and then an average is computed. This ubiquitously-used approach works well, but it requires numerous trials. This in turn causes subject fatigue and, therefore, limits the number of conditions that can be tested for a given subject.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 578 – 585, 2006.

In this paper, a practical denoising technique of the MEG data using the wavelet transform is presented with application results. The remainder of the paper is organized as follows. In Section 2, practical MEG technique and the associated noise problem is discussed in details. Section 3 provides a brief review of the wavelet transform. Section 4 discusses about the denoising technique using the wavelet transform along with the application results, and conclusion is given in Section 5.

2 MEG Technique and Noise Problem

MEG technique measures the extremely weak magnetic field (of the order of femto Tesla, 1 fT = 10^{-15} Tesla) generated by the intracellular neuronal current flow in the brain. This was initiated by the first recordings of the human magnetic alpha rhythm by Cohen in 1968 [2].

The spontaneous or evoked magnetic fields emanating from the brain induce a current in some induction coils, which in turn produce a magnetic field in a special device called a superconducting quantum interference device (SQUID) [3]. The MEG sensors consist of a flux transformer coupled to a SQUID, which amplifies the weak extracranial magnetic field and transforms it into a voltage. Present-day whole-head MEG devices typically contain 64-306 sensors for clinical and experimental works. Overall, MEG technique provides high resolution measurement both in space (2-3 mm) and time (1 ms).

Different techniques have been proposed for analysis of the noisy MEG signals, like, independent component analysis [4], maximum-likelihood technique [5], blind source separation [6] etc. In this paper, we present the wavelet transform-based practical denoising technique of the MEG signals.

The experimental setup used in this work consisted of 274 sensors detecting the magnetic field (fT) for pre- and post-stimulus period, while the stimulus is presented to the subject at time $t = 0$ ms. The total duration of the recording of the sensor data for each trial is for 361 ms, of which 120 ms is for pre- and 241 ms is for post-stimulus period. We are interested for the analysis of the post-stimulus period. 10 trials of the MEG recorded signals using the above-mentioned experimental setup have been used for the experimentation.

3 Wavelet Transform

The Wavelet transform (WT) is a mathematical tool, like the Fourier transform for signal analysis. A wavelet is an oscillatory waveform of effectively limited duration that has an average value of zero. Fourier analysis consists of breaking up a signal into sine waves of various frequencies. Similarly, wavelet analysis is the breaking up of a signal into shifted and scaled versions of the original (or mother) wavelet. While detail mathematical descriptions of WT can be referred to in [7], [8], a brief mathematical summary of WT is provided in the following sections.

The continuous wavelet transform (CWT) is defined as the sum over all time of the signal multiplied by scaled and shifted versions of the wavelet function ψ . The CWT of a signal $x(t)$ is defined as

$$CWT(a,b) = \int_{-\infty}^{\infty} x(t)\psi_{a,b}^{*}(t)dt \ , \tag{1}$$

where

$$\psi_{a,b}(t) = |a|^{-1/2} \ \psi((t-b)/a) \, . \tag{2}$$

$\psi(t)$ is the *mother* wavelet, the asterisk in (1) denotes a complex conjugate, and $a, b \in R, a \neq 0,$ (R is a real continuous number system) are the *scaling* and *shifting* parameters respectively. $|a|^{-1/2}$ is the normalization value of $\psi_{a,b}(t)$ so that if $\psi(t)$ has a unit length, then its scaled version $\psi_{a,b}(t)$ also has a unit length.

Instead of continuous scaling and shifting, the mother wavelet maybe scaled and shifted discretely by choosing $a = a_0^m, b = na_0^m b_0, t = kT$ in (1) & (2), where $T = 1.0$ and $k, m, n \in Z,$ (Z is the set of positive integers). Then, the discrete wavelet transform (DWT) is given by

$$DWT(m,n) = a_0^{-m/2}\left(\sum x[k]\psi^{*}[(k - na_0^m b_0)/a_0^m]\right). \tag{3}$$

By careful selection of a_0 and b_0, the family of scaled and shifted mother wavelets constitutes an orthonormal basis. With this choice of a_0 and b_0, there exists a novel algorithm, known as *multiresolution signal decomposition* [9] technique, to decompose a signal into scales with different time and frequency resolution. The MSD [9] technique decomposes a given signal into its detailed and smoothed versions. MSD technique can be realized with the cascaded *Quadrature Mirror Filter* (QMF) [10] banks. A QMF pair consists of two finite impulse response filters, one being a low-pass filter (LPF) and the other a high-pass filter (HPF).

4 Denoising Using Wavelet Transform

For denoising purpose, first all the 274 sensor recordings (for the post-stimulus period) are concatenated as a single vector of size 1x66034 (66034=274x241). This is followed by denoising using the wavelet transform. MSD [9] approach is used, for 8 scales, using different mother wavelets. This results in (2^8 =)256 times less samples. So, we get an estimate for 66034/256= 258 sensor data. For the rest of the sensors, i.e. 274-258=16, are estimated from the recordings as the mean. These are concatenated with the estimated 256 data from the wavelet analysis to get the 274 sensor data estimation. The final output variable (denoised MEG signal) is constructed by iterating for the 241 post-stimulus period using the denoised estimation. This approach can be applied to get the denoised signal for single representative trial, or for *n* number of trials (iteratively) followed by the average. Obviously the single trial estimation is faster, but the *n*-trial estimation results in better signal quality. If the MSD *N*-scale decomposition results in less number of sensor data (like the case here), we have to perform end-point signal estimation; otherwise if the decomposition results in more number of sensor data, we have to throw away the end-points. We have used three different mother wavelets, Daubechies 4 [7], Coiflets [7] and Adjusted Haar [11]. Fig. 1 shows the average MEG data for the post-stimulus period.

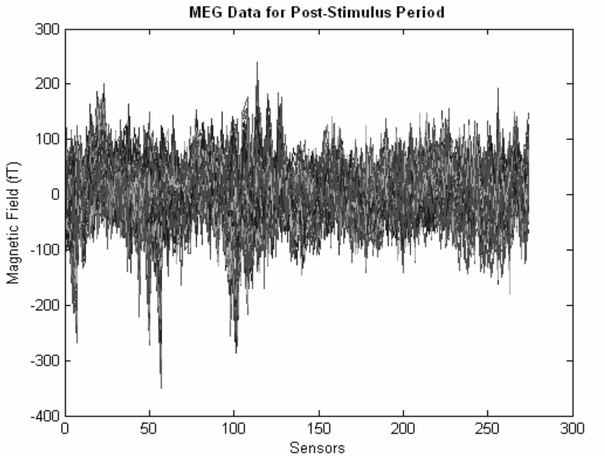


Fig. 1. Average MEG Signal over 10 Trials for 274 Sensors

4.1 Analysis Using Daubechies 4 Mother Wavelet

For the Daubechies 4 [7] wavelet, the scaling function $\phi(x)$ has the form

$$\phi(x) = c_0\phi(2x) + c_1\phi(2x-1) + c_2\phi(2x-2) + c_3\phi(2x-3) \tag{4}$$

where

$$c_0 = (1+\sqrt{3})/4,\ c_1 = (3+\sqrt{3})/4\ ,\ c_2 = (3-\sqrt{3})/4\ ,\ c_4 = (1-\sqrt{3})/4\ . \tag{5}$$

The Daubechies 4 wavelet function $\psi(x)$ for the four-coefficient scaling function is given by

$$\psi(x) = -c_3\phi(2x) + c_2\phi(2x-1) - c_1\phi(2x-2) + c_0\phi(2x-3)\,. \tag{6}$$

We used the daubechies 4 (db4) mother wavelet for the 8-scale signal decomposition and denoising. The end-point estimation was done by iterating over the 10 trials. Fig. 2 shows the denoised signal using the db4 mother wavelet compared against the noisy signal in Fig. 1. The magnitude of the magnetic field (Y-axis) remains more or less at same scale while reducing the superimposed noisy components.

4.2 Analysis Using Coiflet 1 Mother Wavelet

Coiflets are compactly supported symmetrical wavelets [7]. It has orthonormal wavelet bases with vanishing moments not only for the wavelet function ψ, but also for the scaling function ϕ. For coiflets, the goal is to find ψ, ϕ so that

$$\int x^l \psi(x)dx = 0\,, \quad l = 0,1,...,L-1 \tag{7}$$

$$\int \phi(x)dx = 1\,, \quad \int x^l \phi(x)dx = 0\,, \quad l = 0,1,...,L-1\,. \tag{8}$$

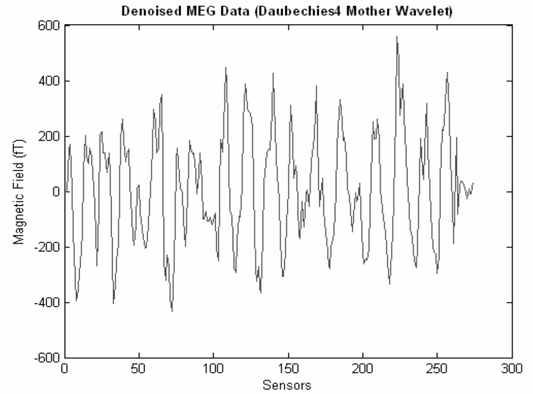


Fig. 2. Denoised MEG Signal using the Daubechies 4 Mother Wavelet

L is called the *order* of the coiflet [7]. Following several tests, we have chosen *L*=1 for our application, which provided the best denoising performance. The 8-scale signal denoising is followed by the end-point estimation by iterating over the 10 trials. Fig. 3 shows the denoised signal using the coiflet 1 mother wavelet.

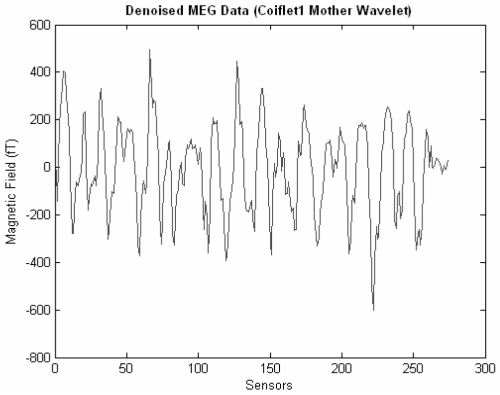


Fig. 3. Denoised MEG Signal using the Coiflet 1 Mother Wavelet

4.3 Analysis Using Adjusted Haar Mother Wavelet

In general, the FIR (finite impulse response) scaling filter for the Haar wavelet is $h = 0.5[1 \quad 1]$, where 0.5 is the normalization factor. As an adjustment and improvement of the characteristics of the Haar wavelet, Ukil & Zivavovic proposed to introduce $2n$ zeroes (n is a positive integer) in the Haar wavelet scaling filter, keeping the first and last coefficients 1 [11]. The scaling filter kernel for the adjustment parameter n is shown below.

$$\begin{aligned} h &= 0.5[1 \quad 1] && \text{for } n = 0 \\ h &= 0.5[1\ 0\ 0\ 1] && \text{for } n = 1 \\ h &= 0.5[1\ 0\ 0\ 0\ 0\ 1] && \text{for } n = 2 \end{aligned} \tag{9}$$

It should to be noted that the original Haar wavelet scaling filter corresponds to $n = 0$, and complex conjugate pairs of zeroes for each $n > 0$ are introduced [11].

It has been shown mathematically in [11] that the introduction of the adjusting zeroes does not violate the key wavelet properties like compact support, orthogonality and perfect reconstruction. A theorem has been proven in [11] which states:
"The introduction of the $2n$ adjusting zeroes to the Haar wavelet scaling filter improves the frequency characteristics of the adjusted wavelet function by an order of $2n$+1."

Following the proof, the adjusted wavelet function $\psi_n(\omega)$ of the adjusted Haar wavelet becomes,

$$|\psi_n(\omega)| = \frac{\{\sin((2n+1)\,\omega/4)\}^2}{|(2n+1)\,\omega/4|} < \frac{4}{|(2n+1)\omega|}\,. \tag{10}$$

The factor $2n$+1 in the denominator of (18) improves the frequency characteristics of the adjusted Haar wavelet function, by decreasing the ripples (as $n > 0$) [11].

We used the adjusted Haar mother wavelet with 4 adjusting zeros for the 8-scale signal denoising. Four zeros were chosen for best possible performance without hampering the speed. Fig. 4 shows the denoised signal using the adjusted Haar wavelet.

4.4 Performance

The performance metric used is the signal-to-interference/noise ratio,

$$Output\ SNIR \cong 10\log_{10}\left(\frac{1}{K}\sum_{i=1}^{K}\frac{\sum_{n=1}^{N} Y_{mean}^2}{\sum_{n=1}^{N}\left(Y_{mean} - Y_{calc}\right)^2}\right)(dB)\,, \tag{11}$$

where $N = 241$ (post-stimulus period), $K = 274$ (no. of sensors), Y_{mean} is the average MEG signal computed over the 10 trials (shown in Fig. 1), and Y_{calc} is the denoised MEG signal using the three different mother wavelets. The output SNIR, indicated as dB, for the denoising operation using the daubechies 4, coiflet 1 and adjusted Haar mother wavelets are -28 dB, -30 dB and -26 dB respectively. Higher values of the output SNIR indicate better performance. Hence, the denoising operation using the adjusted Haar mother wavelet performs best followed by the daubechies 4 and the coiflet 1 mother wavelets.

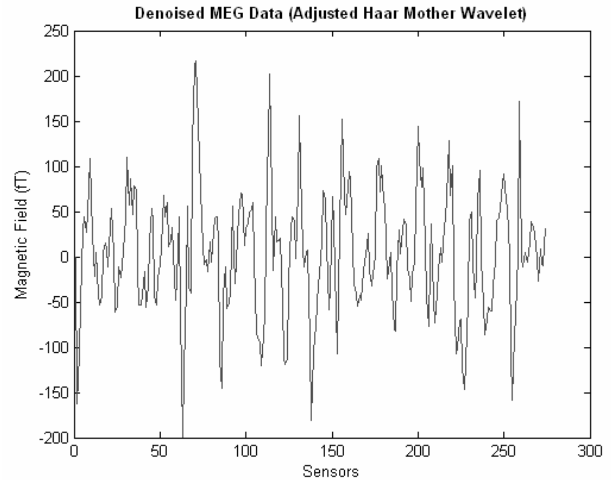


Fig. 4. Denoised MEG Signal using the Adjusted Haar Mother Wavelet

The average computation time using the MATLAB® Wavelet toolbox in an Intel® Celeron® 1.9 GHz, 256 MB RAM notebook was 13.42 s, 14.85 s and 13.64 s respectively for the daubechies 4, coiflet 1 and adjusted Haar mother wavelets.

5 Conclusion

MEG, the noninvasive technique to measure the magnetic fields resulting from intracellular neuronal current flow, is quite important for functional brain imaging. However, the level of noise that is inherent in the data collection process is large enough that it oftentimes obscures the signal(s) of interest. Normal averaging over numerous trials of signal recording most often does not produce optimum result and also causes subject fatigue. In this paper, we have presented the wavelet transform-based denoising technique of the MEG signal. The concatenated MEG signal from 274 sensors is denoised using the mutiresolution signal decomposition technique. Three different mother wavelets, namely, daubechies 4, coiflet 1 and adjusted Haar have been used for the analysis. The denoising performance is quite robust. Hence, the wavelet tranform-based denoising technique of the MEG signals is quite effective from practical point of view.

Acknowledgments

The recorded MEG signals were kindly provided by Kenneth E. Hild II, from the Biomagnetic Imaging Lab., Dept. of Radiology, Univ. of California at San Francisco.

References

1. Paetau, R.: Magnetoencephalography in pediatric neuroimaging. Developmental Sciences, Vol. 5. (2002) 361-370
2. Cohen, D: Magnetoencephalography: evidence of magnetic field produced by alpha-rhythm currents. Science, Vol. 164. (1968) 784-786
3. Zimmermann, J.E., Thiene, P. and Harding, J.T.: Design and operation of stable rf-biased superconducting point-contact quantum devices and a note on the properties of perfectly clean metal contacts. Journal of Applied Physics, Vol. 41. (1970) 1572-1580
4. Ikeda, S., and Toyama, K.: Independent component analysis for noisy data-MEG data analysis. Neural Network, Vol. 13. (2000) 1063-1074
5. de Munck, J.C., Bijma, F., Gaura, P., Sieluzycki, C.A., Branco, M.I. and Heethaar, R.M.: A maximum-likelihood estimator for trial-to-trial variations in noisy MEG/EEG data sets. IEEE Transactions Biomedical Engineering, Vol. 51. (2004) 2123–2128
6. Makeig, S., Jung, T.P., Bell, A.J., Ghahremani, D. and Sejnowski, T.J.: Blind separation of auditory event-related brain responses into independent components. Proc Natl Acad Sci USA, Vol. 94. (1997) 10979–10984
7. Daubechies, I: Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics, Philadelphia (1992)
8. Mallat, S: A wavelet tour of signal processing. Academic Press, New York (1998)
9. Mallat, S: A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. IEEE Trans Pattern Analysis and Machine Intelligence, Vol. 11. (1989) 674-693
10. Strang, G. and Nguyen, T.: Wavelets and filter banks. Wellesley-Cambridge Press, Wellesley-MA (1996)
11. Ukil, A. and Zivanovic, R.: Adjusted Haar wavelet for application in the power systems disturbance analysis. Digital Signal Processing. (under review)

Signal Restoration and Parameters' Estimation of Ionic Single-Channel Based on HMM-SR Algorithm

X.Y. Qiao[1,2], G. Li[1], and L. Lin[1]

[1] Biomedical Engineering Department, Tianjin University,
300072, Tianjin, China
ligang59@tju.edu.cn
[2] Electronic & Information Technology Department, Shanxi University,
030006, Taiyuan, China
xyqiao@sxu.edu.cn

Abstract. Single ion-channel signal of cell membrane is a stochastic ionic current in the order of picoampere (pA). Because of the weakness of the signal, the background noise always dominates in the patch-clamp recordings. The threshold detector is traditionally used to denoise and restore the ionic single channel currents. However, this method cannot work satisfactorily when signal-to-noise ratio is lower. A new approach based on hidden Markov model (HMM) is presented to restore ionic single-channel currents and estimate model parameters under white background noise. In the study, a global optimization method of HMM parameters based on stochastic relaxation (SR) algorithm is used to estimate the kinetic parameters of channel. Then, the ideal channel currents are reconstructed applying Viterbi algorithm from the patch-clamp recordings contaminated by noise. The theory and experiments have shown that the method performs effectively under the low signal-to-noise ratio (SNR<5.0) and has fast parameter convergence, high restoration precision and strong noise robusticity.

1 Introduction

Ion channel is a special large protein molecule spanning the membrane of excitable cells. In the protein molecule there exists a pore, which, in certain conformations, keeps open and allows the passage of selected ions along the electrochemical gradient to form ionic currents in the order of picoampere. In the other conformations the pore keeps closed and no currents exist. Respectively we say the channel is open and closed. The stochastic open and closed states of channel are related to the transmembrane voltage, the mechanical pressure and neurotransmitter. The patch-clamp technique can record the ionic currents flowing through single-channel protein molecules [1]. The error for recordings exists due to the weakness of single-channel currents and the effect of background noise. In order to discover the unknown channels and study the kinetic characters of ion channel as well, it is necessary to accurately restore the channel current from patch-clamp recordings. Generally, signal-channel currents are detected by half-amplitude threshold detection [2]. However,

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 586 – 595, 2006.

because of the small magnitude of the unitary current in many channels, signal-to-noise ratio of patch-clamp recordings is low (SNR<5.0). In this case, the method for threshold detection fails completely. The restoration of ionic single-channel signal based on HMM is an effective means of idealizing patch-clamp recordings under strong background noise [3], [4].

An approach based on hidden Markov model is presented to restore ionic single-channel currents and estimate model parameters under white background noise. In this method, a global optimization algorithm based on stochastic relaxation is used to estimate the kinetic parameters of channel firstly. On the basis, the ideal channel current is reconstructed utilizing Viterbi algorithm from patch-clamp data contaminated by noise. The experimental results have shown the effectiveness of this method.

2 HMM Parameter Estimation

2.1 HMM Basic Theory

HMM is a dual stochastic model. One is Markov chain, which, is described by the transition between states with parameters ($\boldsymbol{\pi}$, $\boldsymbol{A}$) and exports a sequence for states. The other one is stochastic process which is described with parameter $\boldsymbol{B}$ and exports an observed sequence. The parameters are detailedly elucidated as follows:

(1) $\boldsymbol{Q}=\{q_1, q_2,\dots q_N\}$ is a state set for Markov chain in which N denotes the number of states. In this paper, it represents the number of channel current amplitude levels. Usually, $N=2$ or 3. s_t denotes the state at time t. $\boldsymbol{S_T}=(s_1,s_2,\dots,s_T)$.
(2) $\boldsymbol{\pi}=(\pi_1\dots\pi_i\dots\pi_N)$ is initial state probability. Where, $\pi_i=P(s_1=q_i)$,$1\le i\le N$
(3) $\boldsymbol{A}=(a_{ij})_{N\times N}$ is state transition probability matrix. Where, $a_{ij}=P(s_{t+1}=q_j|s_t=q_i)$, $i, j=1,2,\dots,N$.
(4)$\boldsymbol{Y_T}=(y_1\dots,y_t\dots,y_T)$ is an observed sequence, which is sampled from patch-clamp recordings by computer in the paper. T is the length of sampling. $1\le t\le T$
(5)$\boldsymbol{B}=(b_j(y_t))$ is probability density matrix of observation value. Where, $b_j(y_t)=P(y_t|s_t=q_i)$, the probability of observed y_t while the state being q_j at time t. $1\le j\le N$

Therefore, hidden Markov model is denoted with a parameter set $\lambda=(\boldsymbol{\pi}, \boldsymbol{A},\boldsymbol{B},\boldsymbol{Q})$. There are three correlative HMM questions when model λ and observed sequence $\boldsymbol{Y_T}$ are known.

(1) Given λ and $\boldsymbol{Y_T}$, seek the probability $P(\boldsymbol{Y_T}|\lambda)$.
(2) Given λ and $\boldsymbol{Y_T}$, seek $r=(r_t(i))$. Where, $r_t(i)= P(s_t=q_i|\boldsymbol{Y_T}, \lambda_i)$. And obtain the most likely state sequence.
(3) Given λ and $\boldsymbol{Y_T}$, reestimate parameter $\lambda^*=(\boldsymbol{\pi}^*,\boldsymbol{A}^*,\boldsymbol{B}^*,\boldsymbol{Q}^*)$ and seek optimal model parameter λ^{ML}, where ML denotes maximum likelihood estimation.

The fundamental methods to solve above three questions are forward-backward algorithm, Viterbi algorithm and Baum-Welch algorithm [5], [6].

In order to get the optimal ionic single-channel state sequence, the parameters of hidden Markov model are estimated by maximizing a prior probability using Baum-Welch reestimation algorithm. The single-channel current is then uncovered as the most likely state sequence by maximizing a posterior probability using the Viterbi algorithm.

2.2 HMM Description on Patch-Clamp Recordings

Ion channel currents appear quantal in nature, transiting in a seemingly random manner between the open and closed, and have the characteristic of “all” or “none”. They are one by one rectangle, with invariable current amplitude and stochastically variable dwelling duration. Though the current signal of single channel has only two current amplitude levels, which respectively correspond to the open and closed of channel, the channel kinetics has multi open or closed states of different mean dwelling durations (corresponding to different open or closed conformations), which take on same open or closed current levels. This is called the “aggregation” of ion channel conformations [3]. The states are connected by certain way, and the transition between states is indicated with transition rate constant matrix ***R***. Admittedly the transition between all states is a first-order, finite state Markov process [4]. Some channels are more complicated, having more than two current amplitudes (conformation class). Due to the aggregation of the channel conformation states and the background noise from patch-clamp system, the Markov feature of state transition cannot be observed directly. Therefore, we adopt HMM to describe the patch-clamp recordings, which are sum of ion channel currents and background noise.

Because ion channel current signals sampled by computer are discrete at time, original Markov processes convert to a discrete Markov chain. Matrix ***R*** (transition rate constant matrix) denotes transition intensity between states of Markov processes, which is denoted with transition probability matrix $\boldsymbol{A} = (a_{ij})_{N\times N}$ in Markov chain. If sampling interval is Δ, matrix ***R*** can be calculated by $\boldsymbol{A} = exp(\boldsymbol{R}\Delta)$ after matrix ***A*** estimated.

Due to strong background noise, it is different to decide the number of current amplitude levels directly from patch-clamp recordings. Say nothing of deciding conformation states of ion channel. Conformation states are determined only by fitting dwell time histogram of current amplitude signals [7]. In the paper, the word “state” can be directly referred to as the current amplitude, and the transition between different states (current amplitudes) can be considered to be Markovian.

2.3 Parameter Estimation Algorithm Based on HMM

Parameters' estimation based on HMM usually adopt Baum-Welch iterative algorithm. To a given observation sequence $\boldsymbol{Y_T}$, make the probability $P(\boldsymbol{Y_T}|\lambda)$ arrive at local maximum by adjusting each parameter of model $\lambda= (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{Q})$.

Supposing a patch-clamp recording sequence $\boldsymbol{Y_T} =(y_1...,y_t...,y_T)$, the probability $P(\boldsymbol{Y_T}|\lambda)$ may be calculated by the forward-backward algorithm.

$$\mathrm{P}(\boldsymbol{Y_T} \mid \lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) \tag{1}$$

Where, forward variables $\alpha_t(i)$ and backward variables $\beta_t(i)$ respectively are

$$\alpha_1(i)=\pi_i b_j(y_1) \qquad \alpha_t(i)=\sum_{i=1}^{N}\alpha_{t-1}(i)b_j(y_t)a_{ij} \quad 1\le t\le T \tag{2}$$

$$\beta_T(i)=1 \qquad \beta_t(i)=\sum_{i=1}^{N}\beta_{t+1}(i)b_j(y_{t+1})a_{ij} \quad 2\le t\le T-1 \tag{3}$$

To avoid "underflow" phenomena in calculation, we adopt the method to add proportion factor [8]. For forward variables $\alpha_t(j)$:

$$\alpha_1(i)=\pi_i b_j(y_1) \qquad \alpha_1^*(i)=\alpha_1(i)\Big/\sum_{i=1}^{N}\alpha_1(i) \tag{4}$$

$$\alpha_{t+1}^{\#}(j)=[\sum_{i=1}^{N}\alpha_t^*(i)a_{ij}]b_j(y_{t+1}) \quad 1\le j\le N, 1\le t\le T-1 \tag{5}$$

$$\alpha_{t+1}^*(j)=\alpha_{t+1}^{\#}(j)\Big/\sum_{j=1}^{N}\alpha_{t+1}^{\#}(j) \tag{6}$$

Similarly for backward variables $\beta_t(i)$:

$$\beta_T(i)=1 \qquad \beta_T^*(i)=1 \qquad \beta_t^{\#}(i)=\sum_{j=1}^{N}a_{ij}b_j(y_{t+1})\beta_{t+1}^*(j) \tag{7}$$

$$\beta_t^*(i)=\beta_t^{\#}(i)\Big/\sum_{j=1}^{N}\alpha_{t+1}^{\#}(j) \qquad 1\le i\le N, 1\le t\le T-1 \tag{8}$$

Then, estimate kinetic parameter $\lambda^*=(\boldsymbol{\pi}^*, \boldsymbol{A}^*, \boldsymbol{B}^*, \boldsymbol{Q}^*)$ in terms of reestimation formula to make probability $P(\boldsymbol{Y_T}|\lambda^*)$ maximum. Baum-Welch reestimation formula is as follows and its deduction sees also reference literature [9].

Let,

$$\xi_t(i,j)=P(s_t=q_i, s_{t+1}=q_j|Y_T,\lambda),\ r_t(i)=P(s_t=q_i|Y_T,\lambda), \tag{9}$$

$$h_T=(h_T(i))=P(q_i\,|\,Y_T,\lambda) \tag{10}$$

According to Byes rule and Markov characteristics of channel signal, exist

$$\xi_t(i,j)=\frac{\alpha_t^*(i)a_{ij}b_j(y_{t+1})\beta_{t+1}^*(j)}{\sum_{i=1}^{N}\alpha_t^*(i)\beta_t^*(i)} \qquad r_t(i)=\frac{\alpha_t^*(i)\beta_t^*(i)}{\sum_{i=1}^{N}\alpha_t^*(i)\beta_t^*(i)} \tag{11}$$

$$h_T(i)=\frac{1}{T}\sum_{t=1}^{T}r_t(i) \tag{12}$$

Thereby,

$$\pi_i^* = r_1(i) \qquad a_{ij}^* = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} r_t(i)} \qquad q_i^* = \frac{\sum_{t=1}^{T-1} r_t(i) y_t}{\sum_{t-1}^{T-1} r_t(i)} \tag{13}$$

According to obtained parameter $\boldsymbol{Q}^*$ by reestimation formula, parameter $\boldsymbol{B}^*$ is revised correspondingly.

$$b_i^*(y_t) = P(y_t | s_t = q_i^*) = \frac{1}{\sigma_w \sqrt{2\pi}} \exp(\frac{-(y_t - q_i^*)^2}{2\sigma_w^2}) \tag{14}$$

2.4 Algorithm Improving

Baum has proved that the model parameters revised by above reestimation formula satisfy $P(\boldsymbol{Y_T} | \lambda^*) \geq P(Y_T | \lambda)$. However, Baum-Welch iterative algorithm makes $P(\boldsymbol{Y_T} | \lambda)$ local maximum by recursion, not but whole maximum [10]. Therefore, the last parameter values are correlative to choose for parameter initial values. If the initial model parameters inadequately choose, Baum-Welch algorithm usually obtains local optimal solution [11].

In the paper, we use a global optimization algorithm of HMM parameters based on stochastic relaxation (HMM-SR). Namely, by adding a tiny stochastic perturbation, the parameters' training avoids getting into local optimization. Because the effect of HMM parameter $\boldsymbol{A}$ and $\boldsymbol{\pi}$ to objective function is less than that of parameter $\boldsymbol{B}$, we introduce stochastic perturbation only to parameter $\boldsymbol{B}$. Furthermore, training HMM parameters being an iterative process, the perturbation to parameter $\boldsymbol{B}$ will bring a change of forward-backward variables and indirectly influence parameter $\boldsymbol{A}$. SR algorithm is described as follows:

(1) Set the initial HMM parameters, initial probability matrix $\boldsymbol{\pi}$ =[1,0,… 0].

(2) Set maximal iterative times I and convergent condition ξ (for example ξ=10^{-3}).

(3) Let calculating pace m=0.

(4) Set temperature specification T_m=T_0 * $f(m)$. Where, $f(m)$ is a descending function of variable m.
$f(m)$=K^m (K<1)

(5) Produce N×T (N denotes state numbers and T denotes sequence length) independently normal stochastic variable x whose mean is zero and variance is T_m. Let

$$b_i^*(y_t) = b_i(y_t) + x \qquad 1 \leq i \leq \mathrm{N}, \quad 1 \leq \mathrm{t} \leq \mathrm{T} \tag{15}$$

To $b_i^*(y_t)$ unitary processing, we can obtain

$$b_i(y_t) = \frac{b_i^*(y_t)}{\sum_{t=1}^{T} b_i^*(y_t)} \tag{16}$$

(6) If $m>I$ or satisfying convergent condition, end the parameter training. Otherwise, go to (4) for continued training.

On the algorithm, the initial temperature is very important. If it is too low, the global searching ability is restricted. If it is too high, the algorithm is easy to get into stochastic operation at beginning and add training time. The temperature coefficient K value directly influence the degressive speed of system temperature. In the experiment, we choose T_0=1/64 and K=0.98.

3 Statistical Reconstruction Algorithm Based on HMM

That restore current signals from contaminated patch-clamp recordings by statistical technique is to determine the optimal state sequence $s_{1,}\, s_2\ldots\ldots\, s_{T-1}\,, s_T$ according to the given patch-clamp recordings $\boldsymbol{Y_T}$ and estimated model λ. Consider the probability of each state sequence occurring at all time t from 1 to T. The sequence corresponding to probability maximum is to be the channel signal sequence. Namely

$$\mathrm{s}_{1,}\, \mathrm{s}_2\ldots\ldots\, \mathrm{s}_{\mathrm{T}-1}\,, \mathrm{s}_\mathrm{T} = \mathrm{argmax}\ \mathrm{P}(\mathrm{s}_{1,}\, \mathrm{s}_2\ldots\ldots\, \mathrm{s}_{\mathrm{T}-1}\,, \mathrm{s}_\mathrm{T} | \boldsymbol{Y_T}, \lambda)$$

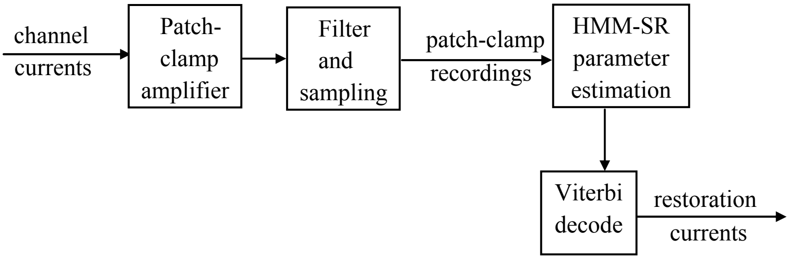


Fig. 1. Reconstruction system based on HMM

We exploit Viterbi algorithm to restore the most likely state sequence after the model parameters have been estimated. To avoid "underflow" questions, logarithmic processing technology is adopted [12]. The algorithm proceeds as follows.

(1) Initializtion: $\delta_1(\mathrm{i})=log[\pi_i]+log[b_i(y_1)]$, $\varphi_1(\mathrm{i})=0$, $1\leq \mathrm{i}\leq N$, $1\leq t\leq T$

(2) Recursion: $\delta_t(\mathrm{j})=\arg\max_{1\leq i\leq N}\ [\delta_{t-1}(\mathrm{i})+log a_{\mathrm{ij}}]+log[b_j(y_t)]$,

$\varphi_t(\mathrm{j})=\arg\max_{1\leq i\leq N}\ [\delta_{t-1}(\mathrm{i})+log a_{ij}]$, $1\leq \mathrm{j}\leq N,\ 2\leq t\leq T$

(3) End: $P^*(\boldsymbol{Y_T}|\lambda)=\max_{1\leq i\leq N}\ [\delta_\mathrm{T}(\mathrm{i})]$, $s_T^*=\arg\max_{1\leq i\leq N}\ [\delta_\mathrm{T}(\mathrm{i})]$

(4) Reconstructing state sequences: $s_t^*=\varphi_{t+1}(s_{t+1}^*)$, $t=T\text{-}1,T\text{-}2,\ldots 1.$

The reconstruction system of ionic single-channel currents based on HMM is shown in Fig.1.

4 Simulation Experiment and Application

4.1 Simulation Experiment

The data were sampled at 20kHz, and a total of 20000 samples were generated under no open channel activity by patch-clamp EPC-10 amplifier. The time sequence is just background noise, which is approximatively white noise under a low cutoff frequency of filter and A/D sample frequency. Noise mean m_ω = 0.0066pA, variance $\sigma_\omega{}^2$ =0.59, and Gaussian distribution. Namely, $\varphi(\omega_t)=N(0,0.59)$. Thereby, the probability density matrix of the observation $\boldsymbol{B} = (b_j(y_k))$ is known. Make standard deviation of noise equal to 1 by multiplying a coefficient (1.302). We denote noise sequence with $\{\omega_t\}$, which indicates the background noise from patch-clamp recordings having the minimum (-2.365pA) and the maximum (2.831pA).

Stimulate a Markov sequence $\{s_t\}$of 20000 samples, which was generated from a two-state model with current amplitude levels 1pA and 0pA, as shown in Fig. 2A (only shown 2000 samples). State transition probability $a_{11}=a_{22}$=0.96, $a_{12}=a_{21}$=0.04, T=20000, N=2, Q=(0pA,1pA), SNR=1.0. Patch-clamp recordings $\{y_t\}$ was simulated by noise $\{\omega_t\}$ superposing to signal $\{s_t\}$ (shown in Fig. 2B). Its maximum is 3.126pA and minimum is -2.508pA.

Supposing that initial state transition probability a_{11}= a_{22} =0.60, $a_{12}=a_{21}$=0.40, $\pi_1=\pi_2$=0.5, $\alpha_t^*(i)$ and $\beta_t^*(i)$ were calculated utilizing forward-backward algorithm. Then, calculate $A^*$and π^* by Baum-Welch reestimation formula and HMM-SR algorithm. Finally, the ideal current amplitude sequence { s_t^* } was reconstructed by Viterbi algorithm. The result is shown in Fig. 2C. The algorithm converges by 16 times iteration, and a_{11}=0.9581, a_{22}=0.9593. ER denotes error rate, which is defined the ratio to the samples restored falsely and the length T of sampling sequence. ER=4.16%. The error mainly appears at the samples which signal $\{s_t\}$ sharply change from 0pA to 1pA or contrarily.

4.2 Application to Practical Data

Under effect of 50 μM GABA receptor agonist, K^+ channel currents were recorded by a cell-attached mode in rat hippocampal neurons of 10-14 days. In these patches, only a single channel was active. Pipette solution (in mmol/L): KCl,120; $CaCl_2$,1;$MgCl_2$,2;HEPES,10;EGTA,10. The depolarizing voltage was -40mV. Before joining GABA, the sampled time sequence was a background noise due to no opened channels. After joining GABA 2 minutes, the data were recorded by patch-clamp EPC-10 amplifier. The data were digitized at sampling rate of 20kHz and low-pass filtered to 5kHz. A sequence for 20000 samples was obtained, which current amplitudes were from minimum -2.87pA to maximum 3.16pA as shown in Fig. 3A (only shown 2000 samples).

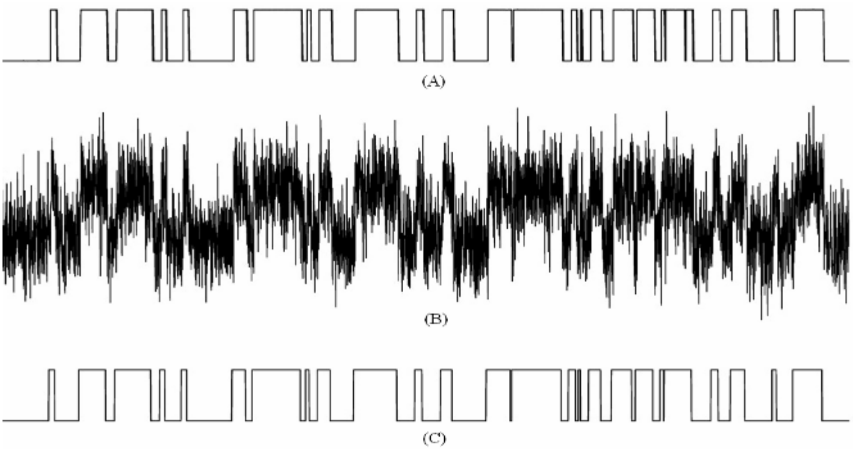


Fig. 2. Simulation results of reconstructing channel signal based on HMM-SR (A) A simulative Markov sequence $\{s_t\}$ (B) A simulative sequence for patch-clamp recordings $\{y_t\}$, (C) A reconstructed current sequence

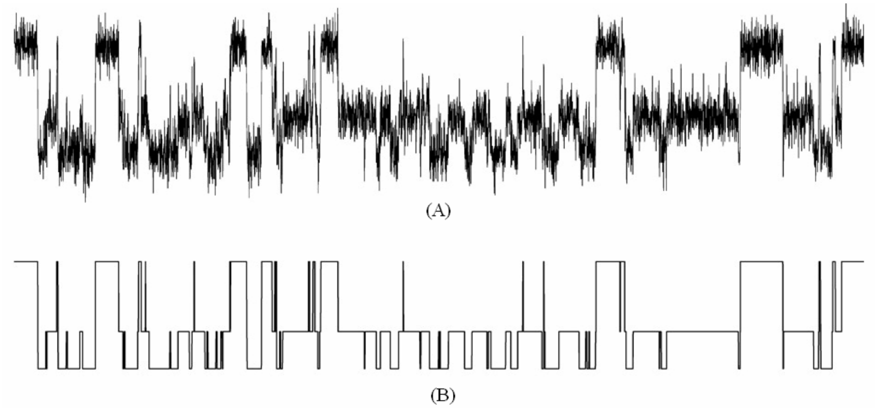


Fig. 3. Practical results of reconstructing channel signal based on HMM-SR (A) Practical data of patch-clamp recordings (B) A restoration sequence by HMM-SR algorithm

According to the sampling sequence, we presume that the channel currents have thirty initial amplitude levels from -2.5pA to 3.3pA for 0.2pA intervals. And transition probability a_{11}=a_{22}=…=a_{29}=0.71, a_{ij}=0.01($i \neq j$). First, calculating $h_T = (h_T(m))$ by Baum-Welch algorithm, the potential current amplitude levels are located at -1pA, 0pA and 2pA after 35 times iteration. The probability distribution for different current amplitude levels is such as Fig. 4, which shows three distinct peaks. Namely, the channel has three open and closed states. Then, the kinetic parameters such as transition probability of ion channel can be estimated by above HMM-SR algorithm, which are convergent to a_{11}=0.9857, a_{22}=0.9662, a_{33}=0.6831 by 18 times iteration. Simultaneously, three current amplitude levels are accurately adjusted to -0.967pA, 0pA and 1.978pA. At last, Reconstruct the ideal channel current amplitude sequence by logarithmic Viterbi algorithm. The result is shown in Fig. 3B. After data idealization, the conformation states can be determined by fitting a current amplitude dwell time histogram.

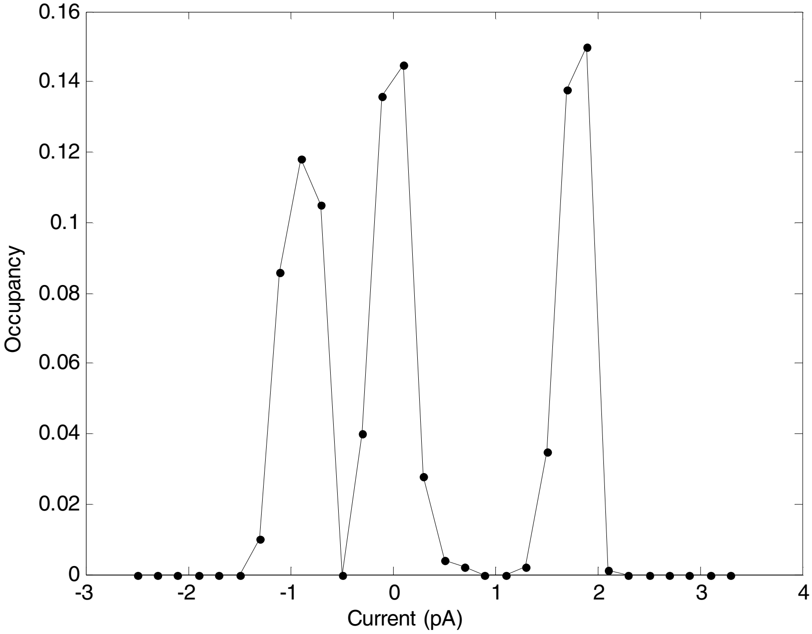


Fig. 4. Probability distribution for different current amplitude levels

5 Conclusions

In this paper, an algorithm based on HMM-SR is applied to effectively solve ion-channel parameters' estimation and signal reconstruction in the patch-clamp technique under white background noise (SNR<5.0). This model makes fully use of the capability to HMM modeling time sequence as well as the global optimization performance of stochastic relaxation algorithm to avoid parameters' training getting into local optimization. The hybrid algorithm has shown the fast convergence, high restoration precision, and strong noise robusticity. Therefore, it can be used to reconstruct ion single-channel currents under a strong background noise.

References

1. Qin, F., Auerbach, A., Sachs, F.: Estimating Single Channel Kinetic Parameters from Idealized Patch-clamp Data Containing Missed Events. Biophys. J., Vol. 70 (1996) 264-280

2. Venkataramanan, L., Sigworth, F.J.: Applying Hidden Markov Models to the Analysis of Single Ion Channel Activity. Biophys. J., Vol. 82 (2002) 1930-1942
3. Chung, S.H., Moore, J.B., Xia, L.G., Premkumar, Gage L.S.: Characterization of Single Channel Currents Using Digital Signal Processing Techniques Based on Hidden Markov Models. Proc. R. Soc. Lond. B boil. Sci., Vol. 329 (1990) 265-285
4. Qin, F., Auerbach, A., Sachs, F.: Hidden Markov Modeling for Signal Channel Kinetics with Filtering and Correlated Noise. Biophys. J., Vol.79 (2000b) 1928-1944
5. Logothetis, A., Krishmurthy, V.: Expectation Maximization Algorithm for MAP Estimation of Jump Markov Linear Systems. IEEE Trans. on Signal Processing, Vol. 47 (1999) 1456-1468
6. Qin, F., Auerbach, A., Sachs, F.: A Direct Optimization Approach to Hidden Markov Modeling for Single Channel Kinetics. Biophys. J., Vol. 79 (2000a) 1915-1927
7. Qin F., Li L.: Model-based Fitting of Signal-channel Dwell-time Distributions. Biophys. J., Vol. 87 (2004) 1657-1571
8. Li, S.X., Tan, J.F., Wei, G.: A Modified Iterative Algorithm for HMM's Parameters. Journal of Circuits and System, Vol. 3 (1998) 82-85
9. He, Q.H., Lu, Y.Q., Wei, G.: A New Approach for HMM Training. Acta Electronica Sinica, Vol. 28 (2000) 56-59
10. Fang, S.W., Dai, B.Q., Li, X.Y.: A Global Optimization Algorithm for Discrete HMM. Journal of Circuits and Systems, Vol. 5 (2000) 78-81
11. Milescu L.S., Akk G. Sachs F. Maximum Likelihood Estimation of Ion Channel Kinetics from Macroscopic Currents. Biophys. J., Vol. 88 (2005) 2494-2515
12. Wu, X.M., Song, C.X., Wang, B.: Hidden Markov Model Used in Protein Sequence Analysis. J. Biomed. Eng., Vol. 19 (2002) 455-458

Signal Sorting Based on SVC & K-Means Clustering in ESM Systems

Qiang Guo[1], Wanhai Chen[1], Xingzhou Zhang[1], Zheng Li[2], and Di Guan[3]

[1] College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China
[2] Southwest Research Institute of Electronic Equipment, Chengdu 610036, China
[3] Harbin Normal University, Harbin 150080, China
guoqiang292004@163.com

Abstract. As radar signal environments become denser and radar signals become more complex, the task of an ESM operator becomes more difficult. This paper presented a de-interleaving/recognition system of radar pulses based on the combination of SVC and K-means clustering. Compared the conventional de-interleaving system, it can produce more complex and compact clustering boundaries according to the distribution characteristics of data set and has good generalization performance. The simulation experiment result shows that the system can sort efficiently radar signals in the high density and complex pulses environment.

1 Introduction

Radar signal sorting is a key technology in electronic support measures (ESM) system.This paper adopted the theories of SVC, K-Means clustering and information entropy. And it presented a novel joint de-interleaving/recognition system on the basis of the combination of SVC & K-Means with the recognition technology of type-entropy. Compared the conventional deinterleaving system[1,2], the SVC sorting method has broken the limit of setting tolerance in the conventional sorting proceeding, it can produce more complex and compact clustering boundaries according to the distribution characteristics of data set and has good generalization performance. In radar signals sorting, the number of data to be handled is very large. If all the data are be treated as the training samples, it would make the scale of adjacency matrix of SVC clustering algorithm enormous. Then the speed of calculation would be affected. Therefore, we can adopt the de-interleaving method of the joint K-Means and support vector clustering to speed the calculation up.

We make entropy as a measure of electromagnetic signal environment, which benefits to quantify the complexity of it. Type-entropy has the capability of macroscopic analysis on electromagnetic signal environment. The result of clustering sorting can be recognized by type-entropy to assist sorting. Through it, we can adjust the parameters of SVC & K-Means sorting so that it could develop a novel system of radar pulse sequence sorting. The experiment result of radar signal sorting in the novel system is to be obtained by computer simulation.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 596–603, 2006.

2 ESM Data Processing Scheme

The ESM data processing scheme has the structure shown in Fig.1. The block former accumulates pulses from the ESM front end. When a certain number have been accumulated, the block of pulses is submitted to the multiparameter clustering sorter.The block former then starts to accumulate another block. The multi-parameter clustering sorter and the *TOA*-difference histogram deinterleave the pulses in the block into pulse chains. Ideally, each pulse chain will consist of all the pulses in the block which have originated from one emitter, and no other pulses. The de-interleaving process takes place in two stages. Firstly, the multi-parameter clustering sorter splits each block of pulses into a number of batches of pulses. The batches are then processed sequentially by the *TOA*-difference histogram, and split into individual pulse chains. The parameters to be entered in the emitter table are then evaluated for each de-interleaved pulse chain by the pulse chain characterizer.The parameters of the characterized pulse chains are then compared with those in the current emitter table by the emitter table updater.A novel multi-parameter sorter is embedded in the ESM data processing system. It de-interleaves radar pulse sequence in multidimensional attribute space according to the character that signals from single radar have self similarity and the signals from different radars have little similarity. The follow is the detailed introduction.

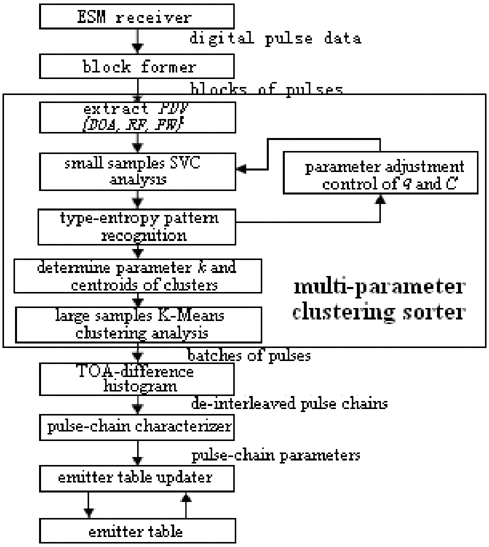

ESM receiver
digital pulse data
block former
blocks of pulses
extract PDV (DOA, RF, PW)
small samples SVC analysis
parameter adjustment control of q and C
type-entropy pattern recognition
determine parameter k and centroids of clusters
large samples K-Means clustering analysis
multi-parameter clustering sorter
batches of pulses
TOA-difference histogram
de-interleaved pulse chains
pulse-chain characterizer
pulse-chain parameters
emitter table updater
emitter table

Fig. 1. Structure of ESM data processing scheme

3 A De-interleaving Method Based on SVC & K-Means Clustering

A radar pulse descriptor vector v_i, $i = 1, 2, \cdots, N$ (N is the length of the radar pulse sequence) with 3-dimension attribute information is constituted with direction of arrival (DOA), radio frequency (RF) and pulse width (PW) of emitter pulse. Sorting algorithm can be summarized in the following steps: Firstly, extract a small sample subsequence whose length is $n(n \ll N)$in the radar pulse sequences and cluster by SVC. And adjust the parameters of SVC to cluster to be k subclass by the type-entropy recognition of complexity. Secondly, figure out the centroid, $G_i(i = 1, \cdots, k)$, of each subclass individually. Finally, treat the number of clusters and centroids at the first step as the initial parameters to clustering sort by K-Means. And we can obtain the final pre-sorting result. This method is described here.

3.1 Support Vector Clustering

Support vector machines [3] are a kind of statistical learning method which is about pre-estimate on finite samples. It founded on the principle of structure risk minimization and combined the idea of the maximal margin classifier with kernel-based learning methods. It shows good generalization performance and can effectively overcome some problems such as the curse of dimensionality, overfitting and so on. At the same time, it can obtain the globally optimal solution. The basic idea of SVC presented by Ben-Hur [4] et al is: first, the data sample is mapped from attribute space to a high dimensional feature space by non-linear transformation. Then we are looking for the optimum separating hypersphere in this new space. The non-linear transformation is founded by kernel function non-linear mapping. We introduce the SVC process on intercepted subsequences here.

Let $V \subseteq R^3$ be a data space of the above radar pulse description vector v_i, with $v_i \subseteq V$, $i = 1, 2, \cdots, N$. The distribution characteristics of received radar pulse parameters are so complex that the boundaries of clusters are also complicate. The clustering feature of the data sets will be more outstanding by using a nonlinear transformation $\varPhi$ from V to some high dimensional feature space. We are looking for the smallest closed convex sphere of radius R in the feature space. This is described by the constraints:

$$\|\varPhi(V_i) - a\|^2 \leq R^2 + \xi_j (\forall j, \xi_j \geq 0) \tag{1}$$

where $\|\bullet\ \|$ is the Euclidean norm and a is the center of the sphere. Soft constraints are incorporated by adding slack variables ξ_j.

To solve this problem, the Lagrangian is introduced

$$L = R^2 - \sum_j (R^2 + \xi_j - \|\varPhi(V_j) - a\|^2)\beta_j - \sum_j \xi_j \mu_j + C \sum_j \xi_j \tag{2}$$

where $\beta_j \geq 0, \mu_j \geq 0$ are Lagrangian multipliers, C is a penalty factor, and $C\sum_j \xi_j$ is a penalty term.

Under the Karush-Kuhn-Tucker conditions [5], we conclude:

1. A point $\Phi(v_i)$ with $\beta_i = C$ is mapped to the outside of the feature space sphere whose the minimal radius is R. The points as v_i will be called outliers and lie outside of cluster boundaries.
2. A point $\Phi(v_i)$ with $0 < \beta_i < C$ is mapped to the surface of the feature space sphere whose the minimal radius is R. The points as v_i will be called Support vectors (SVs) and lie on cluster boundaries.
3. All other points lie inside cluster boundaries.

Throughout this paper, the Gaussian kernel is used:

$$K(v_i, v_j) = \Phi(v_i) \cdot \Phi(v_j) = e^{-q\|v_i - v_j\|^2} \tag{3}$$

with width parameter q.

The Lagrangian Wolfe dual form W is now written as:

$$W = \sum_j K(v_i, v_j)\beta_j - \sum_{i,j} \beta_i \beta_j K(v_i, v_j) \tag{4}$$

At each point v, the distance of its image is defined in feature space from the center of the sphere:

$$\begin{aligned} R^2(v) &= \|\Phi(v) - a\|^2 = \left\|\Phi(v) - \sum_j \beta_j \Phi(v_j)\right\|^2 \\ &= K(v,v) - 2\sum_j \beta_j K(v_j, v) + \sum_{i,j} \beta_i \beta_j K(v_i, v_j) \end{aligned} \tag{5}$$

The radius of the sphere is: $R = \{R(v_i)|v_i \text{ is a support vector}\}$. The contours that enclose the points in data space are defined by the set $\{v|R(v) = R\}$. SVs lie on the contours, which forms the cluster boundaries of the parameters of single radar.

Cluster assignment: given a pair of data points that belong to different clusters, any path that connects them must exit from the sphere in feature space. Therefore, such a path contains a segment of points y such that $R(y) > R$. This leads to the definition of the adjacency matrix A_{ij} between pairs of points v_i and v_j whose images lie in or on the sphere in feature space:

$$A_{ij} = \begin{cases} 1, R(y) \leq R \\ 0, other \end{cases} \tag{6}$$

Clusters are now defined as the connected components of the graph induced by A. Cluster assignment is to be made again based on the connected components by Depth First Search (DFS).

3.2 K-Means Clustering Sorting Based on Centroids

For the data set $V = \{v_1, \cdots, v_N\}$, where $v_i = \{AOA_i, RF_i, PW_i\}, i = 1, \cdots, N$, K-Means will find a partition of V, $P_k = \{C_1, \cdots, C_k\}$, to minimize the value of the target function

$$f(P_k) = \sum_{i=1}^{k} \sum_{v_l \in C_i} d(v_l, m_i) \tag{7}$$

Where $m_i = \frac{1}{n_i} \sum\limits_{v_l \in C_i} v_l$ is the position of the centroid of No.i cluster. $i = 1, \cdots, k, n_i$ is the number of the data items in cluster C_i . $d(v_l, m_i)$ is the distance from v_l to m_i.There are two obvious defects—the initial centroid vector and the number of clusters are given in advance based on the prior information of the data samples distribution—in the radar pulse serial sorting by K-Means clustering method. However, modern electronic countermeasure faces such a radar pulse environment that of complex, dense and insufficient prior information. The "increasing batch" and "missing batch" would be produced significantly when we chose the unsuitable initial centroids and the chosen clustering parameter k isn't the same as the number of real emitters. Therefore, "false alarm" and "false dismissal" are formed unavoidably in the final sorting result [6].

In the de-interleaving method introduced in this paper, firstly, cluster a stage of small samples of radar pulse sequence data by adjusting the clustering parameters of SVC, q and C, to cluster to be k subclass. Secondly, figure out the centroid, $G_i(i = 1, \cdots, k)$, of each subclass individually, i.e. figure out the statistics average value. Finally, treat the above number of clusters, k, and centroids, $G_i(i = 1, \cdots, k)$, as the initial parameters to clustering sort by K-Means. And we can obtain the final pre-sorting result.

The advantages of the joint SVC & K-Means method are:

1. The clustering centroids and the number of clusters, k, are to be set self-adaptively, which is based on the real distribution feature of the data samples instead of the initial centroid and the number of clusters installed in advance by the sorting based on SVC. Moreover, it can avoid the defects of K-Means that the initial centroid is set installed unsuitably so that the iteration of the algorithm is converged to a local optimum.
2. At the same time because of the complexity of SVC algorithm, $O(N^2)$,the data size that of the clustering pulses increased undoubtedly to make a drastic drop in calculating speed. Cluster to the small samples radar pulses data by SVC to get the parameter, k, and the initial centroids of the K-Means clustering by the character of the small samples learning of support vector machines [7]. To process the data sets on a large scale, K-Means algorithm is relatively flexible and efficient because its complexity of time is $O(nkl)$ [8]. Where n is the number of the samples, k is the number of clusters, l is the number of times of iteration when the algorithm is to be converged. Generally, k and l are given in advance, $k << n, l << n$. Therefore, it is a linear relation between the complexity of time of the algorithm and the size of the data sets so that it can meet the need of real time process in radar signal sorting.

4 The Recognition Method of Entropy Measure for Radar Pulses

In electronic countermeasure system, seeking for a suitable physical quantity as the measurement to the complexity of signal environment is not only an urgent need to engineering practice but a difficulty to electronic countermeasure for many years. Adopting the notion of information entropy provided a feasible basis for scientifically evaluating signal environment.

In this paper, radar pulse environment (or its subset) is faced by ESM system is treated as the information source. In this way, complexity of pulse environment can be expressed by uncertainty. In information theory, it is named information source entropy that mean amount of information or uncertainty provided from a message from information source [9]. It is indicated as:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log P(x_i) \tag{8}$$

From (8), we can see that, when the distribution of information source probability space is equiprobability, uncertainty value $H(x)$ is the biggest. Whose size is related to the number of possible states or probability in probability space. The more the number of possible states or the smaller probability, the bigger the uncertainty value is. Information source entropy is the function of probability distribution of information source probability space.

According to the definition of information source entropy, we describe the complexity of signal environment by employing type-entropy in accordance with the character of radar environment. Type-entropy can be indicated as the estimating to the description of pulse categories. Let a pulse be described as RF, PW and DOA, in another word, the same RF, PW and DOA are looked as a kind of pulses, in this sense, we can describe how many the pulse categories are in signal environment by employing type-entropy. It is defined as:

$$H_T(P) = -\sum_{i=1}^{N} P_n \log P_n \tag{9}$$

Where P_n is the probability of each kind of pulses, N is the categories.

5 Parameter Adjustment Control of *Q* and *C* by Type-Entropy

Cluster boundaries are controlled by the width parameter q of Gaussian kernel and the penalty factor C of Lagrangian function in SVC clustering sorting. With parameter q is increased, cluster boundaries perform a more compact character. The size of parameter C determines the number of outliers. With the value C ($C \leq 1$) is reduced, the number of outliers can be increased accordingly.Cluster boundaries can be smoothed by reducing the value C [10].

According to the character that type-entropy value is getting big with the increasing categories and complexity of pulse signals, we can calculate type-entropy on the multi-parameters clustering results. Through recognizing the complexity of it by type-entropy, we can macro-analyze the results of clustering sorting in order to judge it to decide the final parameters q and C of clustering sorting.

6 Simulation Experiment Result

To verify the effect of the novel joint deinterleaving/ recognition system, we adopt the radar signal data as Table 1 in the simulation experiment. At the pulse simulating data being produced, sampling intervals want set and the simultaneously arrived signals are losing proceeded.The first 5000 pulses are sorted in the radar pulse serial data flow. When the above radar pulse sequence is sorted by SVC&K-Means and parameters are adjusted to $q = 30$, $C = 1$ by typeentropy recognition technology, we can obtain a better result of clustering sorting, as Fig.2 and Fig.3. Statistic on the sorting result shows the sorting accuracy is 97.86%.

Table 1. The radar parameters information

radars	PRF	RF	PW	DOA	The number
	(kHz)	(MHz)	(μs)	(deg)	of pulses
Radar1	0.3-0.4	2080-2250	1.2-1.3	48-60	824
Radar2	0.3-0.4	2750-2850	1-1.1	60-80	823
Radar3	0.8-1.0	2250-2350	1.2-1.25	68-80	2149
Radar4	0.7-0.9	2550-2750	1.3-1.4	56-64	1891

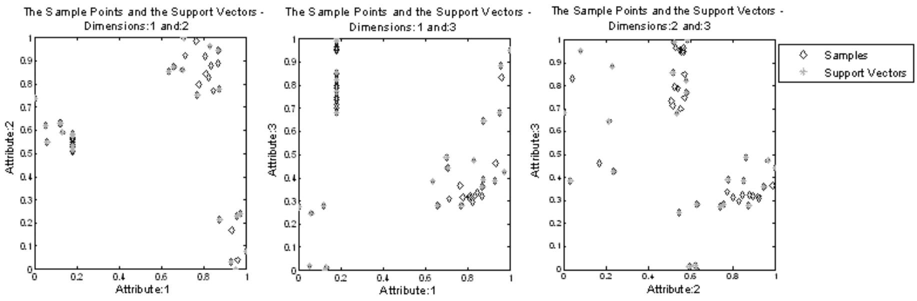


Fig. 2. The distribution of 2-dimension attribute parameters of the clustering result on first 50 data samples by SVC

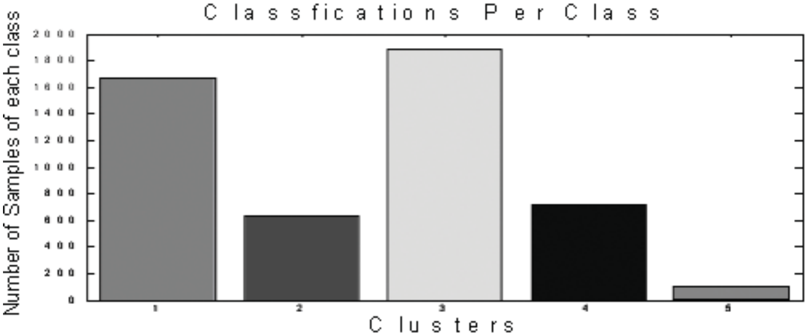


Fig. 3. The statistic histogram of the sorting result by the SVC&K-Means clustering sorter

7 Conclusions

This paper presents a novel joint deinterleaving/ recognition system of radar pulse sequence. It introduces a novel sorting method based on SVC and K-Means clustering. At the same time, the notion of typeentropy is to be adopted and type-entropy recognition is used to assist signal sorting. Simulation experiment shows that the sorting system is effective to the high pulse density environment and the complex signal pattern.

References

1. MILOJEVIC, D.J., POPOVIC, B. M.: Improved Algorithm for De-interleaving of Radar Pulses. IEE Proc. F., Comm., Radar Signal Processing. **139** (1992) 98–104
2. J.A.V. Rogers, Ph.D.: ESM processor system for high pulse density radar environments. IEE Proc. F., Comm., Radar & Signal Processing. **132** (1985) 621–625
3. Cristianini, N., and Shawe-Taylor, J.: An introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press. (2000)
4. Ben-Hur A., Horn D.,Siegel mann H.T. ,and Vapnik V.: Support vector clustering.Journal of Machine Learning Research. **2** (2001) 125–137
5. DENG Nai-yang, TIAN Ying-jie: A new method in data mining: Support Vector Machines. Science Press. (2004)
6. Qiang Guo: A novel multiple-parameter clustering sorting method of radar signal.Journal of Harbin Institute of Technology,unpublished.
7. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition.Data Mining and Knowledge Discovery. **2** (1998) 955–974
8. Mehmed Kantardzic: Data Mining Concepts, Models, Methods, and Algorithms. IEEE Press. (2002)
9. Robert J. McEliece: The theory of information and coding: a mathematical framework for communication, Reading, Mass. Addison-Wesley Pub. Co., Advanced Book Program. (1977)
10. Ben-Hur A., Horn D., Siegelmann H.T., and Vapnik V.: A support vector clustering method. In International Conference on Pattern Recognition. (2000)

Camera Pose Estimation by an Artificial Neural Network

Ryan G. Benton and Chee-hung Henry Chu

Center for Advanced Computer Studies
The University of Louisiana at Lafayette
Lafayette, LA 70504-4330, U.S.A.
{rbenton, cice}@cacs.louisiana.edu

Abstract. Reconstruction of a three-dimensional scene using images taken from two views is possible if the relative pose of the cameras is known. A traditional approach to estimating the pose of the cameras uses eight pairs of corresponding points and involves the solution of a set of homogeneous equations. We propose a multi-layered feedforward network solution. Empirical results demonstrate the feasibility of using the network to recover the relative pose of the cameras in the three-dimensional world.

1 Introduction

Understanding the three-dimensional (3D) world by a computer has such diverse applications as in autonomous vehicle navigation, visualization content creation, surveillance, and digital photogrammetry. Because the 3D world is projected to a 2D image plane, the information loss must be compensated by other means. A passive solution is to use images taken from more than one viewpoint to recover the depth information through the triangulation principle. This has the advantage of not requiring active devices such as ultrasound or lidar sensors, or using intrusive structured lighting, or making specific assumptions about the shape and structures of the scene objects.

The triangulation principle can be used to solve for the 3D location of a scene point if two sets of camera parameters are available, viz. the intrinsic and the extrinsic parameters [1]. The intrinsic parameters are the lens focal length, pixel pitch, and the center of the image plane. These can be obtained off-line via camera calibration methods [4]. The extrinsic parameters refer to the relative orientation and position of the cameras while the two images are taken. Recovery of the extrinsic parameters is often referred to as the pose estimation problem.

The projection of a scene point

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 604–611, 2006.

in homogeneous coordinates on the image plane of a camera is

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

given as:

$$\lambda \mathbf{x} = KPg\mathbf{X},$$

where λ is the distance of the scene point to the camera center, K is the calibration matrix, P is the projection matrix representing the perspective transformation, and g is the transformation that relates the world coordinates to the camera coordinates. The transformation g uses a rotation matrix R and a translation T to move the world coordinates to the camera center. When the camera is calibrated, we can invert K on both sides so that $\mathbf{x}$ is in the normalized coordinates. Since $P = [I \ |0]$, we can re-write the projection equation as

$$\lambda \mathbf{x} = R\mathbf{X} + T,$$

where $\mathbf{X} = [X \ Y \ Z]^T$.

When two images are taken by cameras from different viewpoints, the two corresponding image points are given as [3]:

$$\lambda_1 \mathbf{x}_1 = R_1 \mathbf{X} + T_1,$$

and

$$\lambda_2 \mathbf{x}_2 = R_2 \mathbf{X} + T_2.$$

If we assume the first camera coordinates to be the world coordinates, then we are left with one set of R and T that relates the coordinates of the second camera to those of the first:

$$\lambda_1 \mathbf{x}_1 = \mathbf{X}, \tag{1}$$

and

$$\lambda_2 \mathbf{x}_2 = R\mathbf{X} + T. \tag{2}$$

Since the distances λ_1 and λ_2 are not known, equations (1) and (2) are combined as:

$$\mathbf{x}_2^T \hat{T} R \mathbf{x}_1 = 0, \tag{3}$$

where $\hat{T}$ is the 3×3 skew symmetric matrix so that, for any 3×1 vector v, $\hat{T}v$ is the cross product of T and v.

The unknown 3 by 3 matrix $E = \hat{T}R$ is referred to as the essential matrix. The camera pose information encoded in the matrices $\hat{T}$ and R is recovered as follows. The essential matrix E is first estimated from observed corresponding point pairs extracted from the stereo pair of images; it is then decomposed into its components R and T [2].

The importance of estimating the camera pose can be seen by observing that by substituting Equation (1) into (2), we have

$$\lambda_2 \mathbf{x}_2 = \lambda_1 R \mathbf{x}_1 + T.$$

If R and T are known, we can use the image point locations $\mathbf{x}_1$ and $\mathbf{x}_2$ to solve for either λ_1 or λ_2, which gives the depth corresponding to the respective image point. The individual depth values collectively can be used to constitute the 3D structure information of the scene.

2 Camera Pose Estimation

The essential matrix is usually recovered by an eight-point algorithm, so called because eight pairs of corresponding points $(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)})$, $i = 1, ..., 8$ are used in the estimation. Rewrite Equation (3) in terms of the essential matrix E as

$$\mathbf{x}_2^T E \mathbf{x}_1 = 0, \tag{4}$$

the so-called epipolar constraint for each corresponding point pair $(\mathbf{x}_1, \mathbf{x}_2)$. The unknown matrix E has 9 elements, but the structure of the essential matrix is such that it has at most 8 degrees of freedom, so that it can be determined by 8 pairs of corresponding points. For $i = 1, \cdots, 8$, we can put the ith point pair $(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)})$ in Equation (4):

$$\mathbf{x}_2^{(i)T} E \mathbf{x}_1^{(i)} = 0. \tag{5}$$

We write Equation (5) in terms of the unknown elements of E as

$$\mathbf{a}^{(i)T} \mathbf{e} = 0, \tag{6}$$

where $\mathbf{a}^{(i)}$ is a 9×1 vector whose elements are the pairwise products of the elements of ith point pair $(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)})$ and $\mathbf{e}$ is the 9×1 vector formed by stacking the elements of the matrix E. Equation (6) for $i = 1, \cdots, 8$ can be stacked to form a matrix equation

$$A\mathbf{e} = 0. \tag{7}$$

The linear solution of E from the homogeneous matrix equation (7) requires the singular value decomposition of an 8×9 matrix and is known to be numerically unstable, so that improvements such as bundle adjustments are developed. These add to the considerable computation requirements. Consequently, we explore the use of a multi-layered feedforward neural network to perform pose estimation.

The number of point pairs needed is related to the degrees of freedom in the estimation problem. There are three degrees of freedom in a 3D rotation, and another three degrees of freedom in a 3D translation. In general, we cannot completely recover the translation component, as can be seen from the following. Suppose a scene point is projected onto the image planes of a pair of cameras. If we move the second camera twice as far from the first camera, the same pair of image coordinates would be obtained if we translate the scene point twice as far from the cameras. Hence, we can only obtain the direction of the second camera from the first, and the estimation problem has only five degrees of freedom.

3 Estimation by an Artificial Neural Network

Our hypothesis is that we can train a neural network to learn the camera pose given a set of observed corresponding points. The design issues to be explored include the representations of input data and that of the parameters to be estimated. The input is, in general, a set of matched image point coordinates, and the output are the pose parameters, such as the rotation angles and translation components.

In this work, we follow the convention of using eight pairs of corresponding points and so we are solving an overdetermined system. The input can therefore be the eight point pairs $(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)})$, for $i = 1, \cdots, 8$. Since each image point has two components, we have a total of 32 input values. In a feedforward layered network, our hidden layer is fully connected to the input layer, so that the corresponding point pairs are essentially uncoupled from the network's point of view.

An alternative input representation is to compute the average and difference of the corresponding point pairs. Let the ith point pair be $(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)})$, where

$$\mathbf{x}_1^{(i)} = \begin{bmatrix} x_1^{(i)} \\ y_1^{(i)} \end{bmatrix}$$

and

$$\mathbf{x}_2^{(i)} = \begin{bmatrix} x_2^{(i)} \\ y_2^{(i)} \end{bmatrix}$$

For $i = 1, \cdots, 8$, we compute a pair of average and difference vectors $(\mathbf{a}^{(i)}, \mathbf{d}^{(i)})$, where

$$\mathbf{a}^{(i)} = \begin{bmatrix} (x_1^{(i)} + x_2^{(i)})/2 \\ (y_1^{(i)} + y_2^{(i)})/2 \end{bmatrix}$$

and

$$\mathbf{d}^{(i)} = \begin{bmatrix} x_1^{(i)} - x_2^{(i)} \\ y_1^{(i)} - y_2^{(i)} \end{bmatrix} .$$

The pair $(\mathbf{a}^{(i)}, \mathbf{d}^{(i)})$ couples the ith corresponding point pair $(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)})$. We refer to the 32 values of $(\mathbf{a}^{(i)}, \mathbf{d}^{(i)})$, $i = 1, \cdots, 8$, as the coupled input set.

As discussed before, the pose parameters are in general 5 angles that define the orientation and the location of the second camara relative to the first. These five parameters can each take on a range $[-\pi, \pi]$. Our convention is that the x and y directions are aligned with the horizontal and vertical directions of the image plane. Each camera is looking towards negative infinity along the z-axis. There is no theoretical restriction on the pose of the second camera relative to the first, but clearly there are practical reasons to impose some constraints. For instance, if the headings of the two cameras are π radians apart, they cannot be observing the same scene. In practice, we can also reasonably determine which of the cameras is to the left of the other, so that we can restrict the direction of the translation component.

We decompose the rotation matrix into individual rotations so that

$$R = R_z(\theta_z)R_y(\theta_y)R_x(\theta_x),$$

where θ_z, θ_y, and θ_x are the rotation angles about the z-, y-, and x-axes, respectively. In our work, we restrict each of these angles to the range $[-\pi/4, \pi/4]$. We assume the stereo pair is left-eye dominant, so that the left camera coordinates form the world coordinates. The translation component can then be written in polar form in terms of ρ as the magnitude of the translation and ϕ, γ specifying the direction of the translation. As observed before, the parameter ρ cannot be recovered from the essential matrix. In our work, we restrict γ and ϕ to the range $[\pi/8, \pi/8]$. This is not unreasonably restrictive in typical cases of stereo vision setup. The five parameters to be recovered are therefore θ_x, θ_y, θ_z, ϕ, and γ, within their respective restricted ranges.

Artifical neural networks can be used for functional approximation and for classification problems. Using a functional approximation approach, our solution is to train a network so that it can approximate the parameters as functions of the inputs, viz. to approximate the nonlinear functions

$$\theta_\alpha(\{(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}) : i = 1, ..., 8\})$$

where $\alpha \epsilon \{x, y, z\}$. When we use coupled input set, the function becomes

$$\theta_\alpha(\{(\mathbf{a}^{(i)}, \mathbf{d}^{(i)}) : i = 1, ..., 8\})$$

where $\alpha \epsilon \{x, y, z\}$.

The functional approximation problem can be transformed into a classification problem by binary coding the output values. This requires that each parameter be quantized into a fixed number of bins; the true parameter value then defines a binary pattern, which is to be learned by the network.

4 Experimental Results

In our preliminary functional approximation experiment, we set ϕ and γ to zero. To generate a sample in a training set, we randomly generate 8 scene points as well as a set of 3 pose parameters. The 8 scene points are then projected onto the two cameras to generate 8 pairs of corresponding image points. The data sample vector then consists of the 32 image point coordinates and the 3 target values. This process is repeated as many times as needed to form a training set.

We used a training set of 200 scene points with each of their 3D locations randomly chosen from $[-1, 1] \times [-1, 1] \times [-9, -11]$. The rotation angles were randomly chosen from $[-\pi/4, \pi/4]$, as discussed earlier. The output values were the rotation angles normalized to the range $[-1, 1]$. The test set consisted of 200 scene points with the rotation angles uniformly spaced in the range $[-\pi/4, \pi/4]$.

A network with 32 inputs, 20 hidden units, and 3 output units was used. The neurons used the tanh function as the activation function. The tanh function has only a short, finite interval in which the output is not clamped to -1 or $+1$. A linear activation function used at the output units might increase the likelihood of the network output to produce intermediate values, thus intuitively it should improve the functional approximation performance. In practice, linear activation functions led to difficulty during learning.

We note that if we use the tanh function as the activation function in the output units, the network is better suited to classification tasks than functional approximation. Nevertheless, we would like to verify to what extent the network can follow the target values. The target and output values of the three rotation angles in the test set are shown in Figures 1 to 3. In these plots, the abscissa is the index of the input set. The input set is ordered so that the target value increases from -1 to 1, corresponding to the angles $-\pi/4$ to $\pi/4$. Whereas the nonlinear nature of the neurons limits the ability of the network to work as a function approximator, the network clearly showed the ability to follow the target from -1 to $+1$ over the range for all three rotation angles.

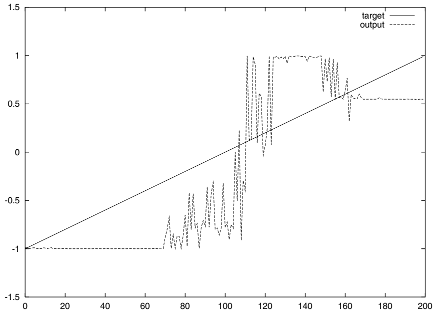


Fig. 1. Target and output values for θ_x

In our second set of experiments, we encode the output values and use the classification capability of the neural network. We quantize a rotation angle to four ranges and use four outputs to represent each of the range. The four ranges were $[-\pi/4, -\pi/8)$, $[-\pi/8, 0)$, $[0, \pi/8)$, and $[\pi/8, \pi/4)$ and the respective target patterns were $[+1, -1, -1, -1]$, $[-1, +1, -1, -1]$, $[-1, -1, +1, -1]$, and $[-1, -1, -1, +1]$. We focus on estimating θ_y, which had the worst performance in our functional approximation experiment. The network has 32 input units, 20 hidden units, and 4 output units.

For these experiments, we use a training set of 2000 samples and a test set of 2000 samples. In the case of uncoupled input data, we were able to obtain a classification rate of 94.85%. In the case of coupled input data, the classification rate was 95.925%.

Four bins form a rather coarse quantization of the range of the rotation angle. If we add more bins, the number of output units in the artificial neural network

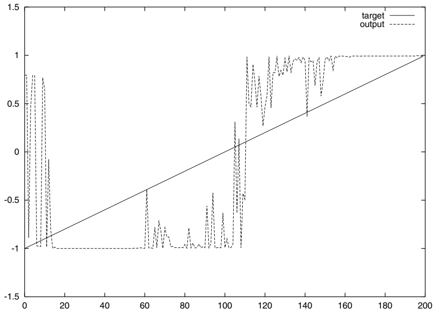


Fig. 2. Target and output values for θ_y

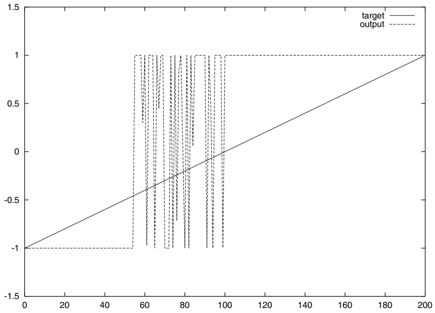


Fig. 3. Target and output values for θ_z

increases. For instance, if we want the rotation angle precision to be within 5 degrees, we need 18 output units. The additional output units may in turn require more hidden units, resulting in a large network that may not be easy to train. It is therefore reasonable to consider using binary coding on the bin index so that, in general, K bins would need only $\log_2 K$ output units. In the example described above, the four ranges could correspond to four patterns, each with two output units: $[-1,-1]$, $[-1,+1]$, $[+1,-1]$, and $[+1,+1]$. The network has 32 input units, 20 hidden units, and 2 output units. In our experiment using uncoupled data, the classification rate drops to 71.5% (57 errors in 200 input sets) in the case of θ_y.

5 Concluding Remarks

Pose estimation is an important step in 3D reconstruction using stereo pairs of images. We train an artificial neural network using a 32-element input representing eight corresponding xy-pairs for each camera pose. The trained network is then tested with novel sets of scene points as well as camera poses. We propose that by configuring the network as a classifier, we can learn the input-output map so as to recover the quantized rotation angles.

Besides demonstrations of the efficacy of this approach, our ongoing work is to optimize the network architecture, to determine the effect of the number of point pairs in the training set, to involve further testing with a wider range of parameters, as well as to quantitatively compare the results with those obtained via the conventional eight-point algorithm.

References

1. R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, Cambridge, 2003.
2. T.S. Huang and O. Faugeras, Some properties of the E matrix in two-view motion estimation. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, pp. 1310-1312, 1989.
3. Y. Ma, S. Soatto, J. Kosecka, and S.S. Sastry, An Invitation to 3-D Vision, Springer, New York, 2004.
4. Z. Zhang, A Flexible New Technique for Camera Calibration, Microsoft Technical Report MSR-TR-98-81, 1998.

Depth Perception of the Surfaces in Occluded Scenic Images*

Baoquan Song[1], Zhengzhi Wang[2], and Xin Zhang[2]

[1] College of Electronic Science and Engineering, National University of Defense Technology, Changsha, 410073, China
baoquansong@gmail.com
[2] Department of Automatic Control, National University of Defense Technology, Changsha, 410073, China

Abstract. The adaptive disparity filter is refined, along with the monocular feature filter designed based on neuron dynamics, and a new stereopsis model is set up in this study. The disparities of the matched binocular features in stereo image pair are detected by the adaptive disparity filter, and removed by the monocular feature filter, only leaving those unmatched monocular features to be added to all depth planes determined by the disparities of the matched binocular features. Finally, visible surface perception is generated by the closed boundaries in the corresponding depth plane during the filling-in processing. By the above mechanism, the depth perception of surfaces in the occluded scenic images is realized, also, the figure-ground segmentation.

Keywords: Neuron Dynamics, Stereo Vision, Occlude, Disparity Filter, Monocular Feature Filter, FACADE Theory.

1 Introduction

Stereoscopic vision is an important research area, so far, most studies on this subject are focused on how to match the image features effectively. In order to reduce the difficulties in features matching, certain constraints have been implemented by the traditional vision theory which is dominated by Marr's computational system of stereo vision (see [1]), however, it doesn't work well in the images of the real-world scenes since these constraints may not be fully complied with (e.g. disparity smoothness constraint). When occlusion occurs, the occluded features in one image have no matchable features in the other image. Besides, if there's no apparent gray scale difference between an occluded area and a non-occluded one, the corresponding features also can not been matched. Not all these problems can be solved by simply improving the feature matching algorithms.

Inspired by the uniqueness and superiority of neural mechanisms such as competition and cooperation, the stereopsis is explored from the other point of view in this study – adopting the achievements in visual physiology and neuropsychology, to develop computer vision system by simulating mechanisms of human

* Supported by Natural Science Foundation of China (60471003, 60334010, 60475029).

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 612–621, 2006.

biological vision. Many researchers developed vision system by analyzing and simulating mechanisms of biological vision (see [2], [3], etc.), and Grossberg's FACADE (Form-And-Color-And-DEpth) theory may be one of the most successful and integrated framework among all the efforts and work in this field, which is based on the neuron dynamics and simulates the functions of the biologic vision system (see [4], [5], etc.). In 2003, Grossberg and Howe refined the FACADE model, set up the 3D LAMINART model (see [5]) to cope with data about perceptual development, learning, grouping and attention. Then, the 3D LAMINART model was developed by Grossberg and Yazdanbakhsh in 2005 to explain how the visual cortex generates 3D percepts of stratification, transparency and neon color spreading in response to 2D pictures and 3D scenes (see [6]). Cao and Grosserg combined FACADE figure-ground mechanisms and 3D LAMINART stereopsis mechanisms, in 2005, proposed the enhanced 3D LAMINART model to explain an even wider range of data about 3D vision and figure-ground perception than was previously possible (see [7]).

FACADE theory has established primary frameworks of binocular vision system upon biological foundations. However, it is chiefly applied to explain some visual phenomena such as illusions and the experimental results by adopting the biological vision mechanism. Both the FACADE model and the 3D LAMINART model can only simulate the visual information processing of simple images which contain regular geometries, and it is difficult for such models to analyze and process images of the real-world scenes in which there are lots of confusions, fuzzy edges.

To implement a practical computational model and construct an image processing computer system that can process real-world images of 3D scenes according to biological mechanism, in 2004, we designed an adaptive disparity filter based on neuron dynamics which implemented the binocular matching of disparity features in stereo images, and integrated it with FACADE theory to achieve a stereopsis system to process complex images with feasible computational load (see [8]). By using this stereopsis system, depth perception of surfaces in real-world stereo images is effectively and efficiently realized, which is not achieved by other models that instantiate the FACADE theory.

However, the stereopsis system proposed in 2004 by us could not process the the occluded scenic images. So, we developed it further. A new stereopsis model is proposed in this study, which could not only achieve depth perception of the surfaces in occluded scenic images but also separate the object from background, resolving those inextricable problems being difficult to the features matching algorithms and the figure-ground segmentation algorithms. In what follows, this stereopsis model and its primary mechanisms are introduced.

2 Stereopsis Model

The diagram of this stereopsis model proposed by us is shown in Fig.1. As what can be seen in it, the monocular preprocessing (MP) of the left and right image inputs generates parallel signals to the oriented contrast filter (OCF) and brightness capturing units via pathways 1 and 2 respectively. MP functions as the lateral geniculate nucleus (LGN) in the biological vision system, modeling the center-surround interaction of the LGN ON cells and the LGN OFF cells, discounting the illuminant by reducing the overall brightness while reserving the ratio contrast of the local image region.

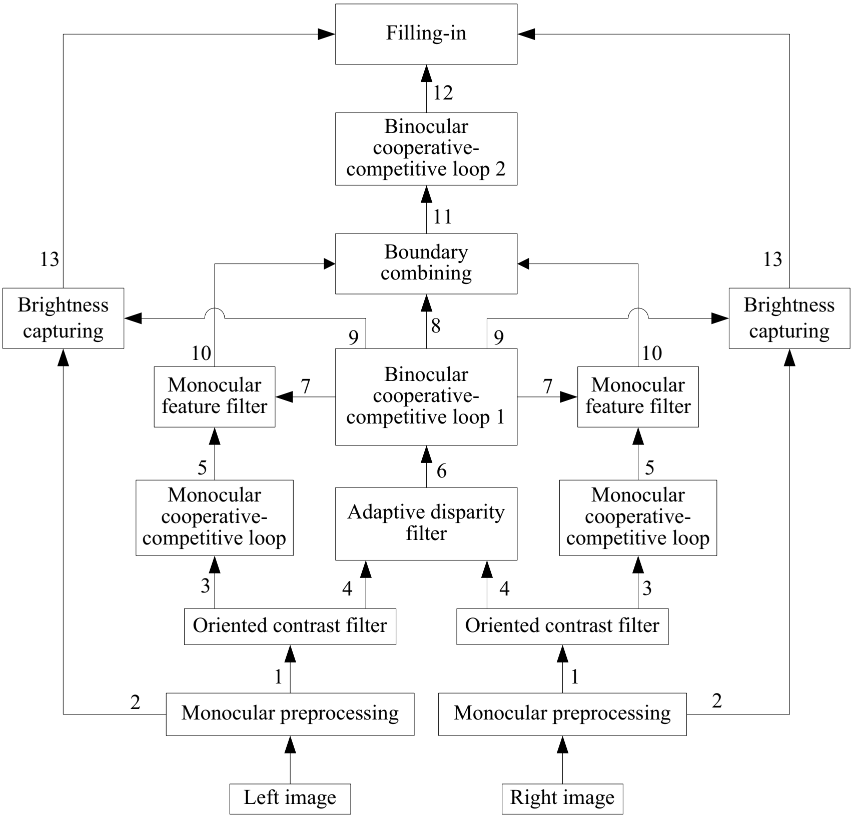


Fig. 1. Diagram of the stereopsis model

The oriented contrast filter models the simple cells in cortical area V1, and fulfills the local contrast detection in certain orientations utilizing a group of contrast detectors in the shape of ellipse. Corresponding to each orientation are pairs of simple cells sensitive to two opposite contrast polarities: one for dark-to-light contrast represented by positive polarity and one for light-to-dark contrast represented by negative polarity. Then, the positive and negative polarities outputs in multi-orientations generated by OCF are transported to the adaptive disparity filter unit via the pathway 4, and the sum of positive and negative polarities outputs in each orientation are transported to the monocular cooperative-competitive loop (CCL) unit via the pathway 3.

The adaptive disparity filter models the binocular complex cells of the cortical area V2. This unit detects the boundary features that can be matched in stereo image pair and forms multiple depth planes to code different disparities. Therefore, It is the key aspect of the stereopsis model. It's outputs reach the binocular CCL 1 via the pathway 6.

Both the monocular CCL and the binocular ones function alike in the model, all consisting of hyper-complex and bi-pole cells, organizing the overall boundary consistently, sharpening and enhancing the fuzzy edges, connecting the broken edges caused by noise, and realizing the boundary grouping and optimizing. In this model, binocular CCL 1 generates parallel signals to the monocular feature filter unit, boundary combining unit and brightness capturing unit via pathways 7, 8 and 9 respectively. Meanwhile, the outputs of monocular CCL are carried to the monocular feature filter unit via the pathway 5.

In the monocular filter unit, the function of monocular complex cells is simulated, and the matched binocular features in stereo image pair are removed, only those unmatched monocular features are left in the left and right images.

The boundary combining unit adds those unmatched monocular features coming from the monocular filter unit via the pathway 10 to all depth planes along their respective lines-of-sight. The outputs of this unit are transported to the binocular CCL 2 via the pathway 11. Through the processing of binocular CCL 2 unit, the grouped and optimized boundaries in each depth plan are carried to the filling-in unit via the pathway 12.

The brightness capturing unit and filling-in unit model the action of the cortical area V4. The brightness capturing unit captures the monocular brightness information from left and right eyes, using the information coming from the pathway 9, forms brightness signals in each depth plane that are transported to the last unit via the pathway 13 to act as the seeds of filling-in operation.

Ultimately, in the filling-in unit, these brightness information and the boundary features are used to fulfill diffusive filling-in within the each depth plane, gaining the depth perception of surfaces and figure-ground segmentation of the real-world scenes.

The following text will only present and discuss those significant processing units of this stereopsis model, full description of the others can be found in [4], [5] and [8], this paper won't repeatedly discuss their details.

2.1 Adaptive Disparity Filter

After MP has discounted the illuminant and OCF has achieved boundary localization, the adaptive disparity filter combines the left and right monocular information, forms multiple depth planes determined by the disparities of the binocular boundary features. The disparities of binocular boundary features in one depth plane are equal, but different depth plane corresponds to different disparity.

The adaptive disparity filter is composed of the binocular combination and the disparity competition. The binocular combination receives the output signals from OCFs of the left and right eyes via the pathway 4. When the complex cells in the binocular combination receive the outputs with approximately the same magnitude of contrast from like-polarity simple cells of left and right eyes, they register a high pattern match and are strongly activated, otherwise they register a less perfect match and aren't strongly activated:

$$F_{ij\hat{k}d} = \max_{-V \leq v \leq V} \left[\left| A^{\mathrm{L}}_{i+v,j,\hat{k},d} + A^{\mathrm{R}}_{i,j,\hat{k},d} \right| \cdot W\left(A^{\mathrm{L}}_{i+v,j,\hat{k},d}, A^{\mathrm{R}}_{i,j,\hat{k},d} \right) \right] \ . \tag{1}$$

$$F_{ij\bar{k}d} = \sum_{v=-V}^{V} \left[\left| A^{\mathrm{L}}_{i,j+v,\bar{k},d} + A^{\mathrm{R}}_{i,j,\bar{k},d} \right| \cdot W\left(A^{\mathrm{L}}_{i,j+v,\bar{k},d}, A^{\mathrm{R}}_{i,j,\bar{k},d} \right) \right] \ . \tag{2}$$

where,

$$A^{\mathrm{L}}_{ijkd} = S^{\mathrm{L},+}_{i+M_d,j,k} - S^{\mathrm{L},-}_{i+M_d,j,k} \ , \ A^{\mathrm{R}}_{ijkd} = S^{\mathrm{R},+}_{i-M_d,j,k} - S^{\mathrm{R},-}_{i-M_d,j,k} \ . \tag{3}$$

In (1), (2) and (3), k represents the different orientation of the contrast, $k \in \{0, 1, \cdots, K-1\}$ (e.g. $k = 0$ represents the horizontal orientation), K is the total number of the orientations, $\bar{k}$ and $\hat{k}$ respectively designates the horizontal and non-horizontal orientations; F_{ijkd} is the total input to the complex cell centered on location (i, j), of orientation k, and tuned to disparity d, it represents the matching degree of the binocular boundary features in every depth plane after processing of the binocular combination; $S^{\mathrm{L/R},+/-}_{ijk}$ denotes the output from the simple cells of the positive and negative polarities in OCFs of the left and right eyes; M_d is the vision shift determined by the disparity d for the binocular combination. $W\left(A^{\mathrm{L}}_{ijkd}, A^{\mathrm{R}}_{ijkd}\right)$ is the weight function revealing the matching degree of the boundary features of left and right images, determined by the adjoining region centered on location (i, j), being used to activate strongly the binocular boundary features with the same polarity and nearly equal contrast.

$$W\left(A^{\mathrm{L}}_{ijkd}, A^{\mathrm{R}}_{ijkd}\right) = \exp\left(-\frac{1}{2\sigma^2} \left(\bar{A}^{\mathrm{L}}_{ijkd} - \bar{A}^{\mathrm{R}}_{ijkd} \right)^2 \right) \ . \tag{4}$$

where,

$$\bar{A}^{\mathrm{L}}_{ijkd} = \sum_{p,q} G_{pqk} \cdot A^{\mathrm{L}}_{i+p,j+q,k,d} \ , \ \bar{A}^{\mathrm{R}}_{ijkd} = \sum_{p,q} G_{pqk} \cdot A^{\mathrm{R}}_{i+p,j+q,k,d} \ . \tag{5}$$

$$G_{pqk} = \beta \exp\left(-\frac{p \cdot \cos\left(\frac{k\pi}{K}\right) - q \cdot \sin\left(\frac{k\pi}{K}\right)}{2\sigma_c^2} - \frac{p \cdot \sin\left(\frac{k\pi}{K}\right) + q \cdot \cos\left(\frac{k\pi}{K}\right)}{2\sigma_s^2} \right) \ . \tag{6}$$

In (5), G_{pqk} is used to reduce the noise effect on binocular combination by Gauss smoothing on the surrounding area; in (6), $\beta > 0$ and $\sigma_c > \sigma_s > 0$.

The max function in (1) and sum function in (2) are used to match the same figure well in the same depth plane, even if it is not the same size in the left and right images, which makes the adaptive disparity filter more robust.

Since the disparities only exist in the matched non-horizontal boundaries that are called as the disparity features in this paper. To the non-disparity features, they are transported directly to the next unit – binocular CCL 1; the phenomenon that the horizontal edges get fuzzy in each depth plan, caused by the sum function in (2), will be reduced by the processing of the binocular CCL 1, for the CCL has the action of sharpening and enhancing the fuzzy edges. While, to the disparity features, the following disparity competition suppresses the false binocular matches in each depth plane, and implements the function that each depth plane only codes the information corresponding its depth.

Because, the activities of complex cells, that have been perfectly activated in the binocular combination, are approximately 2 times as strong as the activities of those that receive the common monocular inputs but aren't perfectly activated. To suppress the false and weak binocular matches in the binocular combination, thus, the dynamics equation of the disparity competition is designed as follows:

$$\frac{d\,J_{ij\hat{k}d}}{d\,t} = -\alpha_1 J_{ij\hat{k}d} + \left(U_1 - J_{ij\hat{k}d}\right) F_{ij\hat{k}d} - \left(J_{ij\hat{k}d} + L_1\right) C \ . \tag{7}$$

where,

$$C = \sum_{e \neq d,p} g\left(F_{i+M_e-M_d+p,j,\hat{k},e}, \Gamma_{ij\hat{k}d}\right) + \sum_{e \neq d,p} g\left(F_{i+M_d-M_e+p,j,\hat{k},e}, \Gamma_{ij\hat{k}d}\right) \ . \tag{8}$$

$$\Gamma_{ij\hat{k}d} = \lambda F_{ij\hat{k}d} \ . \tag{9}$$

$$g(x,y) = \begin{cases} x, & \text{for } x > y \\ 0, & \text{others} \end{cases} \ . \tag{10}$$

In (7), α_1 is a positive constant decay rate, U_1/L_1 bounds the upper or lower limit of cell activity; $J_{ij\hat{k}d}$ is the output activity of complex cell, it represents the final result of the adaptive disparity filter. In (9), $0.5 < \lambda < 1$. Equation (9) and (10) embody the adaptive threshold. Equation (7) has an analytical equilibrium solution, therefore the adaptive disparity filter generates the outputs in one step without iterative operations, which remarkably reduces computational load (see [8]).

2.2 Monocular Feature Filter

The monocular feature filter gets rid of those matched binocular features in the left and right images, leaving those unmatchable features. The neuron dynamics equation is as follows:

$$\frac{d\,X_{ijk}^{\mathrm{L/R}}}{d\,t} = -\alpha_2 X_{ijk}^{\mathrm{L/R}} + \left(U_2 - X_{ijk}^{\mathrm{L/R}}\right) Y_{ijk}^{\mathrm{L/R}} - \left(X_{ijk}^{\mathrm{L/R}} + L_2\right) E^{\mathrm{L/R}} \ . \tag{11}$$

where,

$$E^{\mathrm{L}} = \begin{cases} \sum_{d,v} Z_{i-M_d,j+v,k,d} \,, & \text{for } k = \hat{k} \\ \sum_{d,v} Z_{i-M_d+v,j,k,d} \,, & \text{for } k = \bar{k} \end{cases} \,. \tag{12}$$

$$E^{\mathrm{R}} = \begin{cases} \sum_{d,v} Z_{i+M_d,j+v,k,d} \,, & \text{for } k = \hat{k} \\ \sum_{d,v} Z_{i+M_d+v,j,k,d} \,, & \text{for } k = \bar{k} \end{cases} \,. \tag{13}$$

In (11), (12) and (13), $X_{ijk}^{\mathrm{L/R}}$ is the final output of the left and right monocular feature filter, $Y_{ijk}^{\mathrm{L/R}}$ is the output from the left and right monocular CCL via the pathway 5, $Z_{i,j,k,d}$ is the output from the binocular CCL 1 via the pathway 7.

2.3 Boundary Combining

Since the depth of those unmatched monocular features can't be determined by the adaptive disparity filter, they are added to each depth plane according to (14).

$$H_{ijkd} = Z_{ijkd} + X_{i+M_d,j,k}^{\mathrm{L}} + X_{i-M_d,j,k}^{\mathrm{R}} \,. \tag{14}$$

In (14), H_{ijkd} is the output of the boundary combining.

There are redundant monocular features in each depth plane, however, only those belonging to the object in that depth plane could connect with the binocular features to form the closed boundaries of this object. Finally, in the filling-in unit, only closed boundaries could gain surface perception, while the others don't and disappear.

2.4 Brightness Capturing

The brightness signals are captured by this unit in accordance with (15), using information coming from the pathway 9. Since the disparities only exist in the binocular disparity features, they are used to select the brightness signals that are spatially coincident and orientationally aligned with them in each depth plane, realizing the one-to-many topographic registration of the monocular brightness signals.

$$B_{ijd}^{\mathrm{L},+/-} = I_{i+M_d,j}^{\mathrm{L},+/-} h\left(\sum_{p,k\in\hat{k}} Z_{i+p,j,k,d}\right), B_{ijd}^{\mathrm{R},+/-} = I_{i-M_d,j}^{\mathrm{R},+/-} h\left(\sum_{p,k\in\hat{k}} Z_{i+p,j,k,d}\right) \,. \tag{15}$$

where,

$$h(x) = \begin{cases} 1, & \text{for } x > \Gamma_c \\ 0, & \text{others} \end{cases} \,. \tag{16}$$

In (15), $I_{ij}^{\mathrm{L/R},+/-}$ denotes the LGN ON signal and LGN OFF signal coming from the outputs of the MP unit of left and right eyes via the pathway 2, $B_{ijd}^{\mathrm{L/R},+/-}$ represents the brightness signal outputted by this unit. In (16), Γ_c is the constant threshold and $\Gamma_c > 0$.

2.5 Filling-In

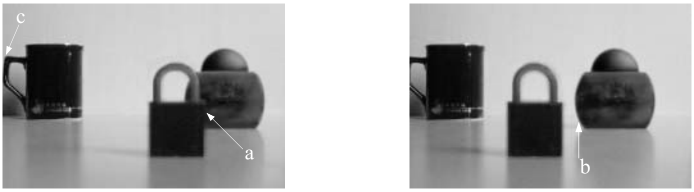
In this unit, firstly the brightness signals from left and right eyes via the pathway 13 are binocularly matched. Within this current implementation, we simply model this matching process as an average of brightness signals of left and right eyes.

$$\tilde{B}^{+/-}_{ijd} = \frac{1}{2}\left(B^{\mathrm{L},+/-}_{ijd} + B^{\mathrm{R},+/-}_{ijd}\right) \ . \tag{17}$$

Then the matched signals $\tilde{B}^{+/-}_{ijd}$ in each depth plane are used for the seeds of filling-in operation, which allows the brightness signals to diffuse spatially across the image except where gated by the presence of boundaries (see [4],[7]). So, only the closed boundaries could gain surface perception by the diffusive filling-in in the corresponding depth plane, while the others don't and disappear, for they can't enclose the brightness signals. Thus, the depth of the unmatched monocular feature is determined by the depth of the matched binocular features, too.

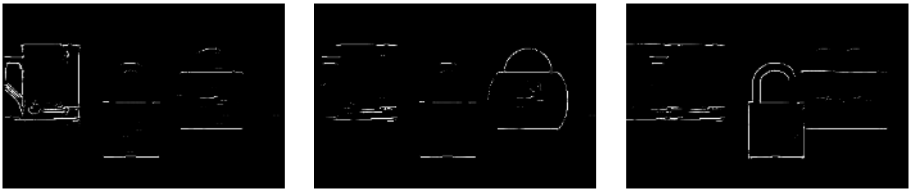
3 Simulation

This model has been realized and simulated in the computer system. Because of space limitations, and to illustrate more expressly, only a sample stereo image pair in which there are three kinds of typically unmatchable monocular features, as presented in Fig.2, are processed and exhibited here. Firstly, the edge feature 'a' between the occluded and non-occluded areas, cannot be matched for lacking of clear gray distinction. Secondly, the monocular feature 'b' in the right image is occluded in the left image, therefore, its corresponding features cannot be



(a) Left image (b) Right image

Fig. 2. The stereo image pair processed in the simulation of this paper



(a) Far plane (b) Middle plane (c) Near plane

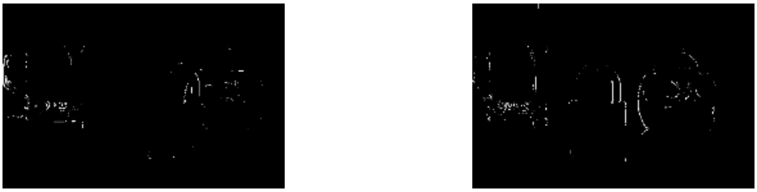
Fig. 3. The matched binocular features

found. Thirdly, the monocular feature 'c' in the left image doesn't appear in the right image. Depth of these three edge features cannot be determined by simply improving the feature-matching algorithm, while it is solved in this model.

Figure 3 shows the final output of the adaptive disparity filter after being processed by the binocular CCL 1. It's easy to see that the binocular features have been matched and their depth planes have been determined. Because of the horizontal bias of eyes configuration in binocular vision system, the horizontal boundary features in these three depth planes are almost the same.

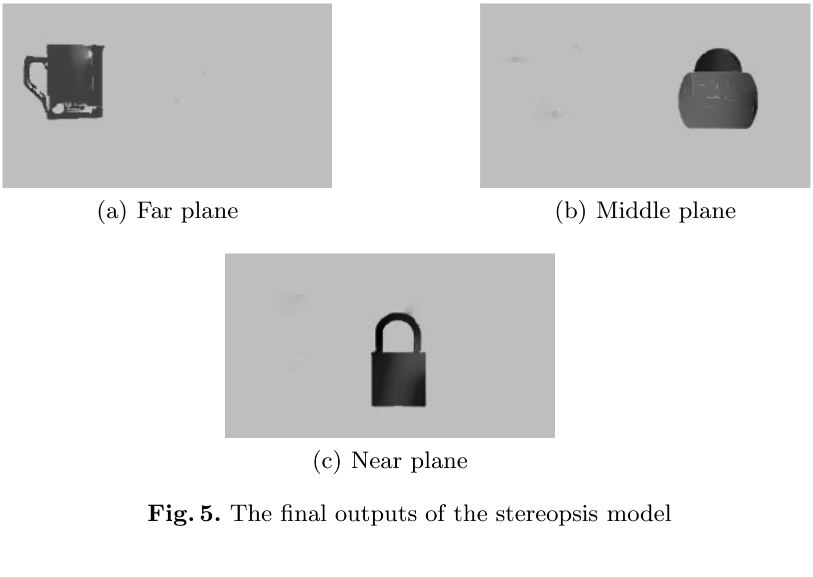
While, the monocular feature filter outputs those unmatched monocular features shown in Fig.4.

The redundant horizontal boundary features and unmatched monocular features in each depth plane will not effect the final 3D surface perception, because those redundant edges that cannot form a closed structure don't produce surface presentation during the final filling-in processing. The final result of this model is shown in Fig.5, it is clear that each depth plane presents an intact figure. The depths of the features in the stereo image pair, including those in the occluded area (viz. feature a and b) and that missed by the camera (viz. feature c), are



(a) The monocular features of left image (b) The monocular features of right image

Fig. 4. The unmatched monocular features



(a) Far plane

(b) Middle plane

(c) Near plane

Fig. 5. The final outputs of the stereopsis model

all determined according to their surface structure. Also, shadows of objects are suppressed and figure-ground separation is fulfilled.

4 Discussion

Stereopsis is researched in this study by applying the biological vision theory in the computer vision, a new stereo vision model is proposed by developing further the adaptive disparity filter and designing a monocular feature filter.

It is proved that the depth perception of the surfaces in the occluded scenes could be gained efficiently by this model. Some difficult problems in the stereo vision field are solved, which cannot be effectively settled by simply improving the feature-matching algorithms and the image segmentation algorithms.

Moreover, a new framework is provided to cope with the occluding problems in the stereopsis. Mathematic depiction of this model is supported by neuron dynamics equations, while actually any mathematic means being capable of implementing the model could be adopted.

References

1. Marr, D.: Vision. 1st edn. W.H.Freeman, San Francisco (1982)
2. Harris, J.M., Parker, A.J.: Independent Neural Mechanisms for Bright and Dark Information in Binocular Stereopsis. Nature. **374** (1995) 808–811
3. Peng, X.H., Zhou, Z.T., et al.: Acquiring Disparity Distribution from Stereo Images with Monocular Cues Based on Cell Cooperation and Competition. Acta Electronica Sinica. **28** (2000) 24-27.
4. Grossberg, S., McLoughlin, N.P.: Cortical Dynamics of Three-Dimensional Surface Perception. Neural Networks. **10** (1997) 1583–1605
5. Grossberg, S., Howe, P.D.L.: A Laminar Cortical Model of Stereopsis and Three-Dimensional Surface Perception. Vision Research. **43** (2003) 801–829
6. Grossberg, S.,Yazdanbakhsh, A.: Laminar Cortical Dynamics of 3D Surface Perception: Stratification, Transparency, and Neon Color Spreading. Vision Research. **45** (2005) 1725-1743.
7. Cao, Y., Grossberg, S.: A Laminar Cortical Model of Stereopsis and 3D Surface Perception: Closure and da Vinci Stereopsis. Spatial Vision. **18** (2005) 515-578.
8. Song, B.Q., Zhou, Z.T., et al.: A New Computational Model of Biological Vision for Stereopsis. In: Yin, F.L., Wang, J., Guo, C.G. (eds.): Advances in Neural Networks - ISNN 2004. Lecure Notes in Computer Science, Vol. 3174. Springer, Berlin Heidelberg (2004) 525-530.

Incremental Learning Method for Unified Camera Calibration

Jianbo Su and Wendong Peng

Department of Automation,Shanghai Jiao Tong University
Shanghai, 200240, P.R. China
{jbsu, pengwendong}@sjtu.edu.cn

Abstract. The camera model could be approximated by a set of linear models defined on a set of local receptive fields regions. Camera calibration could then be a learning procedure to evolve the size and shape of every receptive field as well as parameters of the associated linear model. For a multi-camera system, its unified model is obtained from a fusion procedure integrated with all linear models weighted by their corresponding approximation measurements. The 3-D measurements of the multi-camera vision system are produced from a weighted regression fusion on all receptive fields of cameras. The resultant calibration model of a multi-camera system is expected to have higher accuracy than either of them. Simulation and experiment results illustrate effectiveness and properties of the proposed method. Comparisons with the Tsai's method are also provided to exhibit advantages of the method.

1 Introduction

Camera calibration is to establish a mapping between the camera's 2-D image plane and a 3-D world coordinate system so that a measurement of a 3-D point position can be inferred from its projections in cameras' image frames. In a large variety of applications, multiple cameras are often deployed to construct a stereovision [1] or a multi-camera system [2],[3] in order to provide measurements for 3-D surroundings. For these vision systems, details of a single camera model, i.e., the internal and external parameters of the camera involved, are not important and explicit calibration methods appear to be too fragile and expensive, which leads to the implicit calibration [4]. Most of works in this category are based on neural networks [5],[6],[7], which take advantage of capability of neural networks to approximate a continuous function with arbitrary accuracy. The neural network is trained offline, with the image positions of a feature in cameras as inputs and the 3-D coordinate of the feature as output. This method is simple in methodology, thus has presumably been accepted by those who have to respect a stereovision system but are not expertised in computer vision theory.

However, it is often confusing to use neural network for camera calibration in practice since it is not clear what kind of structure of the neural network should be deployed and which learning algorithm is acknowledged to converge to a reasonable performance. Moreover, if the configuration of a vision system is changed, its neural network-based calibration model should be trained again,

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 622–631, 2006.

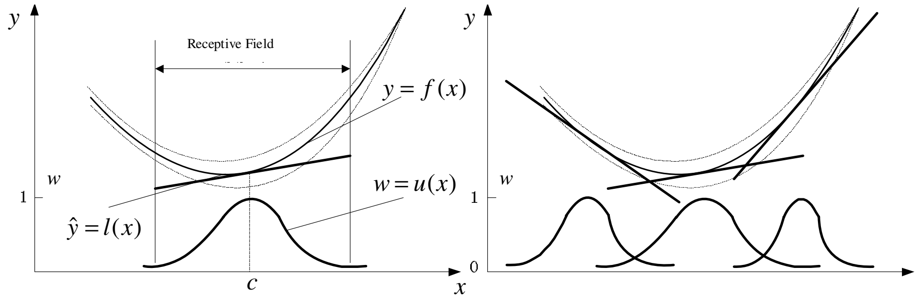
even if most of the cameras in the system are the same. This will increase the cost of the system, especially when reconfiguration of the vision system is frequent.

In this paper, we propose a new implicit method for multi-camera calibration based on receptive fields and data fusion strategy [8]. The receptive field weighted regression (RFWR) algorithm was first proposed by Shaal [9] as an incremental learning method that can overcome negative inference and bias-variance dilemma problems. A nonlinear function is approximated by a set of linear functions, each of which associates with a weighting function describing its approximation accuracy. The definition domain of a linear function is called a receptive field. The final estimation to the nonlinear function is the regression result of all linear functions' estimations weighted by their respective weighting functions. Accordingly, the nonlinear model of a camera can be learned and realized by the RFWR models. The nonlinear mapping between a 3-D position and its 2-D image projections defined by a stereovision system or a multi-camera vision system can be implemented based on the RFWR models of each camera with the help of a weighted average fusion algorithm. The number of receptive fields is evolved automatically according to predefined approximation accuracy. So it is not necessary to determine prior structure of the calibration model or its initial parameters, which facilitates its practical utility to a great extent.

The paper is organized as follows: Section 2 presents preliminaries of receptive field weighted regression algorithm. Section 3 shows how the algorithm is applied to camera calibration problem. Simulation and experiment results of the proposed method are reported in Section 4 and Section 5 respectively, followed by conclusions in Section 6.

2 Receptive Field Weighted Regression

The receptive field weighted regression algorithm is an incremental learning algorithm based on local fields [9]. The essence is to approximate a globally nonlinear function by a number of locally defined linear functions whose definition domains are called receptive fields and each of them is a partition of the definition domain of the nonlinear function. Fig. 1 illustrates principle of the algorithm. In Fig. 1(a), $y = f(x)$ is a nonlinear function to be approximated. The neighborhood region between two dotted curves illustrates noises and other uncertainties that affect approximation to the nonlinear function. The receptive field is a region in x axis, where two functions, $\hat{y} = l(x)$ and $w = u(x)$, are defined. $\hat{y} = l(x)$ is the local linear function to approximate $y = f(x)$, $\hat{y}$ is an estimation to y in this receptive field. $w = u(x)$ indicates approximation effect of $\hat{y}$ to y. For example, $w = u(c) = 1$ means the linear model has a perfect approximation to the nonlinear function at point c. Obviously, a nonlinear function should be jointly approximated by several linear functions defined on different receptive fields, as shown in Fig.1(b). In each receptive field, an output $\hat{y}$ can be obtained from the associated linear model as an estimation to y along with a weight w specifying contribution of $\hat{y}$ to the final approximation of y. Here we call the linear model $\hat{y} = l(x)$ the regression model and $w = u(x)$ the uncertainty model.



(a) A receptive field.

(b) Multiple receptive fields.

Fig. 1. Receptive fields

RFWR algorithm consists of two steps: 1) learning on the receptive field; and 2) generating prediction with weighted average algorithm. For a training sample $(\mathbf{x}, y)$, assuming there are K receptive fields to yield estimations to the true function relations between $\mathbf{x}$ and y. The linear and weighting models in the k-th $(k = 1, \ldots, K)$ receptive field can be expressed as:

$$\hat{y}_k = (\mathbf{x} - \mathbf{c}_k)^T \mathbf{b}_k + b_{0,k} = \tilde{\mathbf{x}}^T \beta_k, \tag{1}$$

$$w_k = exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{c}_k)^T D_k (\mathbf{x} - \mathbf{c}_k)\right). \tag{2}$$

A new receptive field is created if a training sample $(\mathbf{x}, y)$ does not activate any of the existing receptive fields by more than a weight threshold w_{gen}, while a receptive field is pruned if it overlaps another receptive field too much. The overlap is detected when a training sample activates two receptive fields simultaneously more than a predefined weight threshold w_{prun}.

It is clear that the update of $\mathbf{M}_k$ cannot be implemented by using (2) directly. A cost function should be employed, which addresses the final approximation errors of the RFWR model [8]:

$$J = \frac{1}{2}\|y - \hat{y}\|^2 + \frac{\sum\limits_{k=1}^{K} w_k (y - \hat{y}_k)^2}{\sum\limits_{k=1}^{K} w_k}. \tag{3}$$

The first part of Eq. (3) emphasizes on the bias between the actual output y and the estimation from the whole RFWR model $\hat{y}$. Moreover, according to the property of RFWR, each weight w_k is more directly related to the local estimation bias, which leads to the second part of the cost function in (3). This part also leads to a balance among all the receptive fields in terms of local estimation errors in each of individual fields. We believe that balance among all receptive fields is important because this implies approximation accuracy from all receptive fields are similar and estimations from all receptive fields have

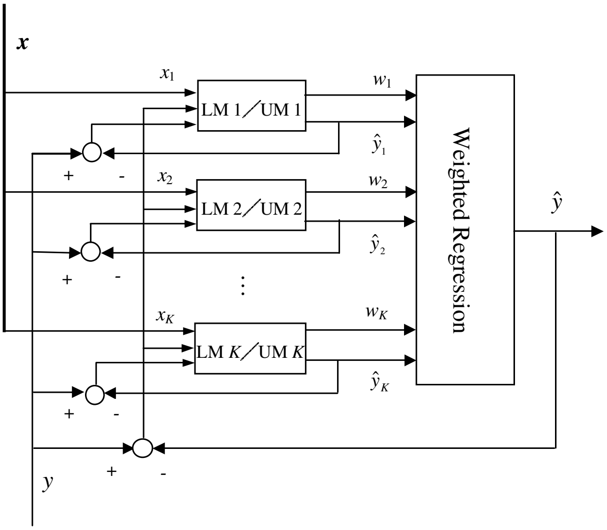


Fig. 2. Learning procedure and the inner structure of an RFWR model. (LM stands for linear model and UM stands for uncertainty model)

identical contributions to the final result. Fig. 2 shows the learning procedure by the cost function in (3), in which LM stands for the linear model and UM stands for uncertainty model.

To learn the uncertainty model in the k-th receptive field, we minimize J with respect to $\mathbf{M}_k$:

$$\frac{\partial J}{\partial \mathbf{M}_k} = (\hat{y} - y)\frac{\hat{y}_k - \hat{y}}{J_1}\frac{\partial w_k}{\partial \mathbf{M}_k} + \frac{(y - \hat{y}_k)^2 - \frac{J_2}{J_1}}{J_1}\frac{\partial w_k}{\partial \mathbf{M}_k}, \tag{4}$$

where we define $J_1 = \sum_{k=1}^{K} w_k$ and $J_2 = \sum_{k=1}^{K} w_k (y - \hat{y}_k)^2$. Eq. (4) is efficient for iterating $\mathbf{M}_k$ in RFWR training.

It is straightforward that the final estimation $\hat{y}$ for a query point $\mathbf{x}$ from all receptive fields comes from a weighted average algorithm [10] that takes all estimations and their associated weights into account:

$$\hat{y} = \frac{\sum_{k=1}^{K} w_k \hat{y}_k}{\sum_{k=1}^{K} w_k}. \tag{5}$$

Fig. 2 also shows this procedure as a part of a complete RFWR model. It is worthwhile noting that the number of the linear models and uncertainty models, thus the number of receptive fields, included finally in a RFWR model is produced automatically according to the learning procedure. Parameters of the linear model defines a local estimation to the global input-output relations, whereas parameters of the uncertainty model defines the shape and size of the receptive field, thus accuracy weights of the local estimation.

3 Implicit Camera Calibration

The mapping from 2-D image feature points and their corresponding 3-D object position is inherently a nonlinear function of the cameras' internal parameters and their relative positions and orientations. We can utilize RFWR algorithm to learn the nonlinear mapping of the vision system and obtain RFWR models for it, which is actually an implicit calibration for the vision system.

We take the known observations from the image planes of the cameras as input and the unknown object position in 3-D space as output. If $\mathbf{T} = (x, y, z)$ is the position of the object in a properly defined 3-D world coordinate system, $\mathbf{C}_i = (u_i, v_i), (i = 1, \ldots, N)$ is the image coordinate of the object from the i-th camera among all N cameras, there exists a nonlinear relation:

$$\mathbf{T} = \mathbf{f}(\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_N). \tag{6}$$

Calibration of the multi-camera system is to learn the nonlinear function $\mathbf{f}$ by RFWR algorithm. Practically, we adopt one RFWR model for each of input-output relations, i.e.

$$\begin{cases} \hat{x}_i = RFWR_x^i(u_i, v_i) \\ \hat{y}_i = RFWR_y^i(u_i, v_i)\,, \qquad i = 1, \ldots, N. \\ \hat{z}_i = RFWR_z^i(u_i, v_i) \end{cases} \tag{7}$$

where $\hat{x}_i, \hat{y}_i$, and $\hat{z}_i$ are estimations from the i-th RFWR model for x, y and z, respectively.

After training, we can obtain N RFWR models for estimations of each of coordinate positions of x, y and z, respectively, i.e., $RFWR_x^i, RFWR_y^i, RFWR_z^i, (i = 1, \ldots, N)$. The final estimation for either of them must be an integrated result from its all RFWR models. For example, the estimation for x should be an integrated results from all $RFWR_x^i, (i = 1, \ldots, N)$ models. It is again natural to employ a weighted average algorithm to fuse estimations from all RFWR model. Suppose there are $K_i, (i = 1, \ldots, N)$ receptive fields in the i-th RFWR model, $RFWR_x^i$, in x direction, then the final estimation for x is

$$\hat{x} = \frac{\sum_{j=1}^{K_1} w_{1j}\hat{x}_{1j} + \sum_{j=1}^{K_2} w_{2j}\hat{x}_{2j} + \cdots + \sum_{j=1}^{K_N} w_{Nj}\hat{x}_{Nj}}{\sum_{j=1}^{K_1} w_{1j} + \sum_{j=1}^{K_2} w_{2j} + \cdots + \sum_{j=1}^{K_N} w_{Nj}}, \tag{8}$$

where w_{ij} and $\hat{x}_{ij} (i = 1, \ldots, N, j = 1, \ldots, K_i)$ are the weight and local estimation at the j-th receptive field of i-th RFWR model $RFWR_x^i$, respectively. Fig. 3 illustrates the structure of the calibration method based on RFWR models and the weighted regression data fusion strategy to elaborate final results with estimations from different RFWR models of different cameras.

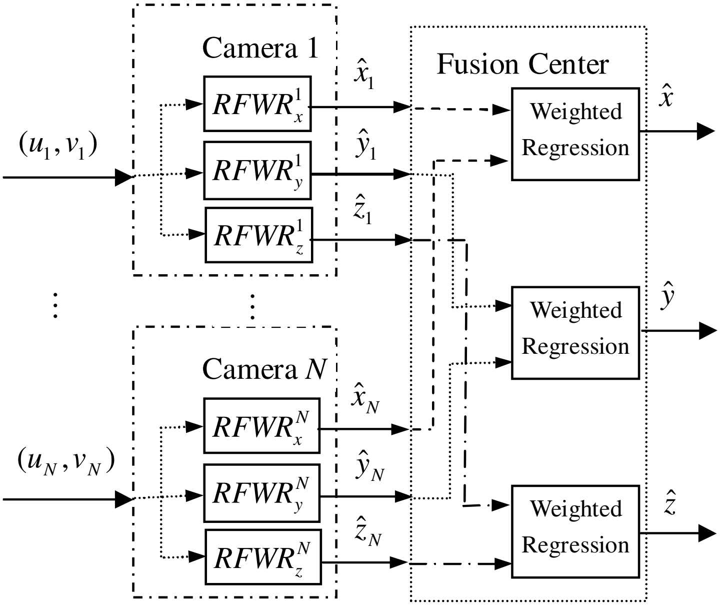


Fig. 3. Camera calibration based on RFWR models and weighted regression algorithms

From the structure of the RFWR model shown in Fig. 2 and the calibration method shown in Fig. 3, we can see that 1) For the fusion algorithm based on RFWR, every receptive field is a subsystem to be fused, with fusion algorithm of weighted regression. Hence, the system shown in Fig. 2 is actually a fusion system composed of several subsystems of similar fusion structures; 2) RFWR algorithm is an incremental learning algorithm. When a new receptive field is generated, it does not affect other existing receptive fields. When the approximation space is enlarged, the whole model is updated by only generating new receptive fields to cover enlarged space. The updated model is not only fit for the enlarged part of the approximation space, but also fit for original part. This incremental learning ability is very important for applications in dynamic environment and dynamic tasks. When a vision system is updated to include more cameras, only the receptive field models of the newly added cameras are included to update the model of the whole vision system. In this sense, RFWR-based calibration method has better adaptability than other implicit calibration methods; 3) RFWR algorithm not only describes an approximate model for the whole vision system, but also provides uncertainty measurements for the approximation.

4 Simulations

We consider a vision system of two cameras. The pinhole model is adopted for all cameras with radial distortion and imaging noises. The internal parameters of each camera are set the same. The focal length is $6mm$, the size of image plane is $6 \times 3mm^2$, and the first-order radial distortion coefficient is 0.01. The imaging noise is a Gaussian type with $N(0, 0.001)$. The two cameras are fixed in a world coordinate system, with similar orientations defined by Euler angles $(90°, -90°, 180°)$. Their positions in the world coordinate system are $(400, -50, 450)$ for camera 1, and $(400, 50, 450)$ for camera 2.

To clarify the calibration procedure, we calibrate the vision system in x, y and z directions independently. That is, an independent calibration model is to be set up for each of the coordinate directions. Without loss of generality, we take the calibration in x direction as an example to demonstrate calibration procedure of the proposed method. Calibrations in y and z directions follow the same procedure described below.

We first generate a number of training data pairs, each data pair composes of a 3-D point in the common visual field of camera 1 and camera 2 and its projections in the cameras' image planes based on their internal and external models. In simulations, we randomly select 90 pairs of sample data for each of the cameras, among which 60 are for training and 30 are for evaluation test. Two RFWR models, $RFWR_x^1$ and $RFWR_x^2$, are respectively established for two cameras in x direction. The convergence condition for training is prescribed that the 3-D reconstruction errors of all training data are less than $1mm$.

The training iterations converge after 4 epochs for $RFWR_x^1$ and 1 epoch for $RFWR_x^2$. Each of the two models includes 15 receptive fields after training. The trained models are fused by weighted average algorithm to obtain final results. The training and fusion results are shown in Table 1, in which ME denotes the maximum error and MSE denotes mean squared error, and they are all measured in millimeters(mm).

Table 1. Training and fusion results for a stereovision system(Unit:mm)

	$RFWR_x^1$	$RFWR_x^2$	Fusion
ME	0.2320	0.2164	0.1756
MSE	0.0052	0.0069	0.0046

From simulations, we found that training procedure for RFWR is stable and easy to converge with high accuracy. Besides, it is clear that the fusion results are better than either of the models, which exhibits good capability of the fusion strategy in suppressing errors from individual RFWR models.

5 Experiments

We apply the RFWR-based method to establish calibration model for a stereovision system that provides visual feedback for robot control. Similar to many other calibration methods [11], a plane of grids is adopted as the calibration reference in the experiment, shown in Fig. 4. The reference plane consists of 7×9 black squares, each of which has the size of $5 \times 5mm^2$.

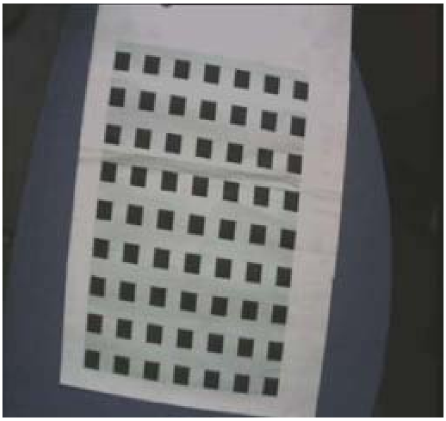


Fig. 4. A planar calibration reference of squares with identical and known sizes

Table 2. Back-projection errors in x direction from RFWR model

Items	$RFWR_x^1$		$RFWR_x^2$	
	Training	Test	Training	Test
ME	3.3274	4.3420	3.8856	4.4406
MSE	0.7689	2.5152	1.1632	2.9102

We take advantage of corner points of all squares as reference points for calibration. Cameras in the stereovision system take images of the reference plane. The corner points in images are extracted with SUSAN algorithm. Due to inconsistency of SUSAN method, all extracted points are preprocessed to reject those false corner points. Consequently, 252 corner points of 63 black squares are obtained.

Coordinates of the 252 corner points in the two camera image planes are known via image processing. In experiment, we simply set up a world coordinate system, with its origin at the lower-left corner point of the calibration reference. So positions of all 252 feature points in the world coordinate system are known. Each corner point's image coordinates and world coordinate are combined to be a sample data vector and all 252 data vectors are used to train the RFWR models of the two-camera vision system. Training procedures converge after 7 iterations, with 121 receptive fields in $RFWR_x^1$ and 140 in $RFWR_x^2$. The training results are evaluated by reconstruction errors of the corner points' positions in 3-D world coordinate system, as shown in Table 3, in which only statistical errors in x direction are listed.

The training errors shown in Table 3 are relatively large due to improper distribution of the sample data. From Fig. 4 , it is easy to see that there are no calibration squares in the edge areas of the image plane. Meanwhile, the cameras used in experiment are of large distortions since the calibration squares in image planes are deformed explicitly.

Table 3. Reconstruction errors by RFWR model and Tsai's model

Items	$RFWR_x^1$	$RFWR_x^2$	Fusion	Tsai
ME	2.8864	3.1606	1.2090	2.5013
MSE	1.9998	2.0158	0.8996	1.6915

The stereovision system is also calibrated by the well known two-step method proposed by Tsai [11] for comparison. We randomly take 16 corner points that can be found in both image planes to train the RFWR-based calibration model for each of the two cameras. Reconstruction errors from both methods are shown in Table 4. Results by RFWR method are expectedly better than those in Table 3. It is clear in Table 4 that although either of the RFWR models of the two cameras has higher error than Tsai's model does, the fusion results from the RFWR models are better than Tsai's model.

Moreover, Fig.5 demonstrates reconstruction differences (Z axis) between Tsai's model and RFWR model in terms of image coordinates (X and Y axes) of camera 1. It is obvious that the reconstruction differences between these two models are small in the central regions and large in the boundary regions. If we take Tsai's model as the ground truth, this means RFWR model has larger errors in boundary regions of the image planes. This observation agrees with distribution of the sampling data for RFWR model training shown in Fig. 4.

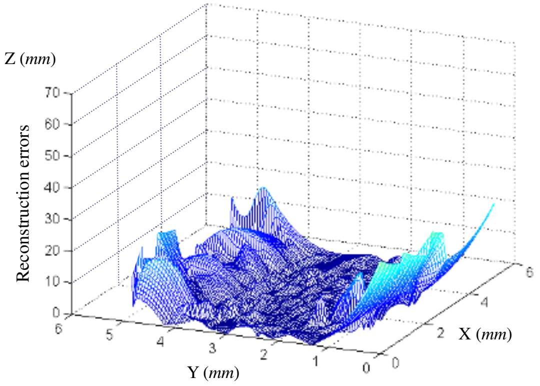


Fig. 5. Reconstruction differences of Tsai's model and RFER models of the stereovision system

6 Conclusion

A new implicit camera calibration method has been presented in this paper based on receptive field weighted regression algorithm. This method inherently approximate the nonlinear camera calibration models with piecewise linear approximations, which are evolved via a learning procedure. Since the approximation accuracy is associated with the set of linear models from learning procedure, final results can easily be obtained by weighted regression fusion algorithms. Moreover, fusion strategy is also exhibited to be an effective way for a multi-camera vision system to achieve better performance. Simulations and experimental results show the performance of the proposed method. Comparisons with Tsai's method are also provided to show its advantages.

References

1. H. Hashimoto, J.H. Lee, and N. Ando, "Self-Identification of Distributed Intelligent Networked Device in Intelligent Space," *Proc. of IEEE Inter. Conf. on Robotics and Automation*, 2003, 4172-4177.
2. F. Pedersini, A. Sarti, and S. Tubaro, "Multi-camera Parameter Tracking," *IEE Proceedings: Vision, Image and Signal Processing, 148*(1), 2001, 70-77.
3. F. Porikli and A. Divakaran, "Multi-Camera Calibration, Object Tracking and Query Generation," *Proc. of Inter. Conf. on Multimedia and Expo*, 2003, 653-656.
4. G.Q. Wei and S.D. Ma, "Implicit and Explicit Camera Calibration: Theory and Experiments," *IEEE Trans. on Pattern Analysis and Machine Intelligence, 16*(5), 1994, 469-480, 1994.
5. M.T. Ahmed and A.A. Farag, " A Neural Optimization Framework for Zoom Lens Camera Calibration," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, 1*, 2000, 403-409.
6. Y.T. Do, "Application of Neural Network for Stereo-Camera Calibration", *Proc. of the IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems*, 1999, 2719-2722.
7. J. Jun and C. Kim, "Robust Camera Calibration Using Neural Network," *Proc. of IEEE Region 10 Conference, 1*, 1999, pp. 694-697.
8. J.B. Su, J. Wang, and Y.G. Xi, "Incremental Learning with Balanced Update on Receptive Fields for Multi-Sensor Data Fusion," *IEEE Trans. on System, Man and Cybernetics, Part B, 34*(1), 2004, 659-664.
9. S. Schaal and C.G. Atkeson, "Constructive Incremental Learning from Only Local Information," *Neural Computation, 10*(8), 1998, 2047-2084.
10. R.A. Jacobs, M.T. Jordan, S.J. Nowlan, and G.E. Hinton, "Adaptive Mixtures of Local Experts," *Neural Computation, 3*, 1991, 79-87.
11. R.Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE J. of Robotics and Automation, 3*(4), 1987, 323-344.

Implicit Camera Calibration by Using Resilient Neural Networks

Pınar Çivicioğlu[1] and Erkan Beşdok[2]

[1] Erciyes University, Civil Aviation School, Avionics Dept., 38039, Kayseri, Turkey
civici@erciyes.edu.tr

[2] Erciyes University, Engineering Faculty, Photogrammetry Devision, 38039, Kayseri, Turkey
ebesdok@erciyes.edu.tr

Abstract. The accuracy of 3D measurements of objects is highly affected by the errors originated from camera calibration. Therefore, camera calibration has been one of the most challenging research fields in the computer vision and photogrammetry recently. In this paper, an Artificial **N**eural Network **B**ased Camera Calibration **M**ethod, **NBM**, is proposed. The **NBM** is especially useful for back-projection in the applications that do not require internal and external camera calibration parameters in addition to the expert knowledge. The **NBM** offers solutions to various camera calibration problems such as calibrating cameras with automated active lenses that are often encountered in computer vision applications. The difference of the **NBM** from the other artificial neural network based back-projection algorithms used in intelligent photogrammetry (photogrammetron) is its ability to support the multiple view geometry. In this paper, a comparison of the proposed method has been made with the **B**undle **B**lock **A**djustment based back-projection algorithm, **BBA**. The performance of accuracy and validity of the **NBM** have been tested and verified over real images by extensive simulations.

1 Introduction

Camera calibration is the process of transforming the 3D position and orientation of the camera frame into 2D image coordinates [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. Camera calibration is a very important step in many machine vision applications such as robotics, computer graphics, virtual reality and 3D vision. The goal of machine vision is to interpret the visible world by inferring 3D properties from 2D images. In this sense, machine vision employs camera calibration in the process of modelling the relationship between the 2D images and the 3D world. In the traditional camera calibration methods such as Bundle Block Adjustment based self-camera calibration (BBA), the coordinate transformation is made using camera calibration parameters that include rotation angles (ω, ϕ, κ), translations (X_0,Y_0, Z_0), the coordinates of principal points (u_0, v_0), scale factors (β_u, β_v) and the skewness (λ) between image axes. In the literature, several methods have been implemented using various camera calibration parameters such as radial lens distortion coefficients (k1, k2), affine

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 632–640, 2006.

image parameters (A, B) and decentering lens parameters (p1, p2). Non-linear optimization algorithms are used to obtain camera calibration parameters in most of the traditional approaches [1, 2, 3, 4, 5, 10, 11, 12, 13, 14, 15].

The Intelligent Photogrammetry (Photogrammetron) [15, 16, 17] consists of ideas, methods and applications from digital photogrammetry, intelligent agents and active vision. Full automation of the photogrammetry, which is referred as intelligent photogrammetry, can only be possible with an autonomous and intelligent agent system. Photogrammetron is basically a stereo photogrammetric system that has a full functionality of photogrammetry in addition to an intelligent agent system and a physical structure of active vision. It may have different forms as coherent stereo photogrammetron, separated stereo photogrammetron and multi-camera network photogrammetron. Photogrammetron can be used in various applications including photogrammetry-enabled robots, intelligent close-range photogrammetry, intelligent video surveillance and real-time digital videogrammetry. The method proposed in this paper, **NBM**, is a novel application of Photogrammetron [15, 16, 17] which provides transformation of 3D world coordinates into 2D image coordinates using an artificial neural network (ANN) structure without restricting the use of zoom lenses and various camera focal-lengths. Therefore, the camera calibration parameters have been described as the ANN parameters such as weights and transfer functions.

In the literature, various camera calibration methods have been implemented with ANNs [3, 4, 11, 12, 14, 15, 16, 17, 18]. Most of these methods employ an ANN structure either to learn the mapping from 3D world to 2D images coordinates, or to improve the performance of other existing methods. Knowing that the camera calibration parameters are important in various computer vision applications such as stereo-reconstruction, the **NBM** goes beyond the existing ones by providing 3D reconstruction from multi-view images besides stereo-images [3, 4, 11, 12, 14, 15, 16, 17, 18].

In this paper, the camera calibration problem is addressed within a multilayer feed-forward neural network (MLFN) structure [19, 20, 21, 22, 23]. The **NBM** can be used with automated active lenses and does not require a good initial guess of classical camera calibration parameters.

The rest of the paper is organized as follows: *A Novel Approach For Camera Calibration Based On Resilient Neural Networks* is explained in Section 2. *Experiments and Statistical Analysis* are given in Section 3. Finally, *Conclusions* are given in Section 4.

2 A Novel Approach for Camera Calibration Based on Resilient Neural Networks

A number of calibration methods employing an ANN structure have been introduced recentlyS [3, 4, 11, 12, 14, 15, 16, 17, 18]. These methods generally require a set of image points with their 3D world coordinates of the control points and the corresponding 2D image coordinates for the learning stage. However, **NBM** offers to use not only stereo images but also multi-view images obtained at

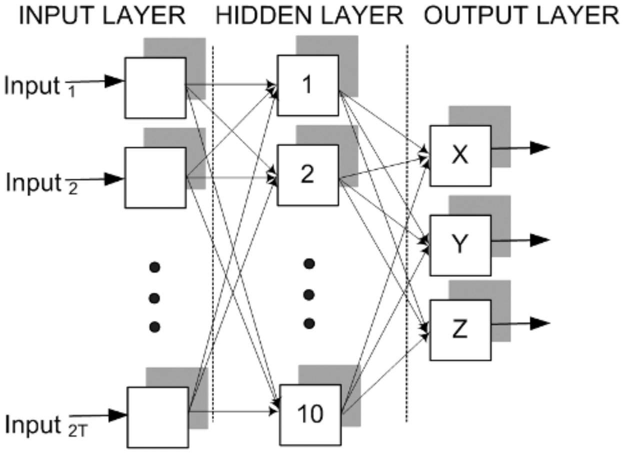


Fig. 1. Structure of the employed MLP

different camera positions as well. This provides using different image scales as if zoom lenses are used. Thus, changes in the geometry of the control points on images help the ANN to learn the relationship between 3D world and 2D image coordinates easily and accurately.

In this paper, the preparation steps of the learning and training data (training_input, training_output and test_input) used in the training of the proposed ANN structure to obtain the 3D world coordinates (X,Y,Z) of a point (p) are given below.

Training data preparation steps:

- Find out the images (j) that encapsulate the point (p) whose 3D world coordinates of (X,Y,Z) will be computed.
- Determine the control points whose image coordinates (u_j,v_j) can be obtained in images (j) found in the first step and obtain the image coordinates (u_j,v_j) of the control points.
- Convert all the image coordinates (u_j,v_j) of ith control point into a raw vector so that Training_Input_i=[u_{ij} v_{ij}], where j denotes the related image number.
- Train the ANN structure using the Training Data (Training_Input, Training_Output).

Training Data are obtained using the images that encapsulate point (p) of whose (X,Y,Z) coordinates will be computed. Training Data designed for each point (p) can have different number of input parameters. Therefore, preparation of Training Data is an adaptive process. Extensive simulations show that this approach increases the accuracy of Back-Projection.

Back-Projection computation steps of the 3D world coordinates (X,Y,Z), of a point (p), using a trained ANN structure are given below:

- Find out the images (j) that encapsulate point (p), whose 3D world coordinates (X,Y,Z) will be computed.
- Find out the image coordinates (u_{pj}, v_{pj}) that belong to the point (p) in these images (j).
- Convert the obtained image coordinates (u_{pj}, v_{pj}) into a raw vector so that Test_Input=[u_{pj} v_{pj}], where j denotes the related image number.
- Simulate the trained ANN structure using Test_Input raw vector in order to compute the (X,Y,Z) coordinates of point (p).

As shown in Fig. 1, the Multi-layer perceptron (MLP) has been used to translate the image coordinates (u,v)$_i$ into the world coordinates (X,Y,Z)$_i$. In the hidden and output layers of the implemented ANN structure, 10 and 3 neurons have been used, respectively. All the neural network structures have been trained with a resilient back-propagation algorithm through a linear transfer function with 1000 epochs. The number of neurons used in the input layer of the ANN structures has been taken as twice the number of images in which the point (i) is encapsulated. For example, the point (i) is encapsulated in 4 images with their (u,v)$_i$ image coordinate pairs and, therefore, the number of neurons in the ANN input layer is taken as T=8 (see Fig. 1).

2.1 Resilient Neural Networks (Rprop)

ANN [22, 23] is an advanced learning and decision-making technology that mimics the working process of a human brain. Various kinds of ANN structures and learning algorithms have been introduced in the literature [20, 21, 22, 23]. In this study, an ANN structure and Rprop learning algorithm have been used [22, 23].

In contrast to other gradient algorithms, this algorithm does not use the magnitude of the gradient. It is a direct adaptation of the weight step based on local gradient sign. The Rprop generally provides faster convergence than most other algorithms [21]. The role of the Rprop is to avoid the bad influence of the size of the partial derivative on the weight update. The size of the weight change is achieved by each weight's update value, $A_{ji}(k)$, on the error function $E(k)$, which is used to calculate the delta weight as in Equation 1.

$$\Delta w_{ji}(k) = \begin{cases} -A_{ji}(k) & if\ B(k) > 0 \\ +A_{ji}(k) & if\ B(k) < 0 \\ 0 & else \end{cases} \tag{1}$$

where $B(k)$ is $\frac{\partial E}{\partial w_{ji}}(k)$ and

$$A_{ji} = \begin{cases} \eta A_{ji}(k-1), & B(k-1)B(k) > 0 \\ \mu A_{ji}(k-1), & B(k-1)B(k) < 0 \\ A_{ji}(k-1), & else \end{cases} \tag{2}$$

where $B(k-1)$ is $\frac{\partial E}{\partial w_{ji}}(k-1)$, η and μ are the increase and decrease factors, respectively where $0 < \mu < 1 < \eta$.

More details about the algorithm can be found in [21, 22, 23].

3 Experiments and Statistical Analysis

A set of real images have been employed in the analysis of the **NBM**. Then the obtained results have been compared with the BBA method. BBA requires the calibration of the camera, therefore, the camera, Casio QV 3000EX/ır, used in the study, has been calibrated.

In the analysis of the **NBM**, the images obtained from the camera have been employed directly and no deformation corrections have made in the images. The

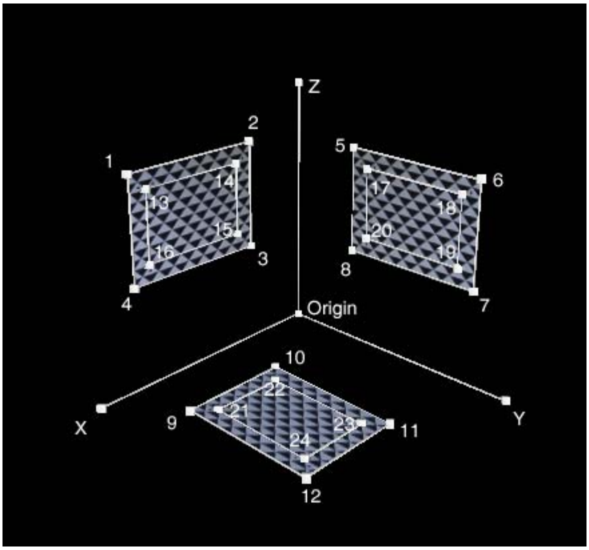


Fig. 2. Positions of the used 24 control points

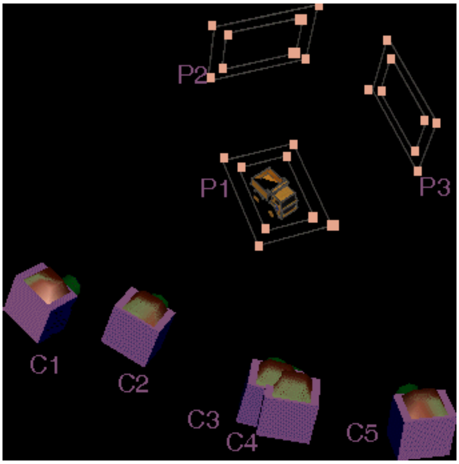


Fig. 3. C(1,2,3,4,5) camera positions and P(1,2,3) control point-planes

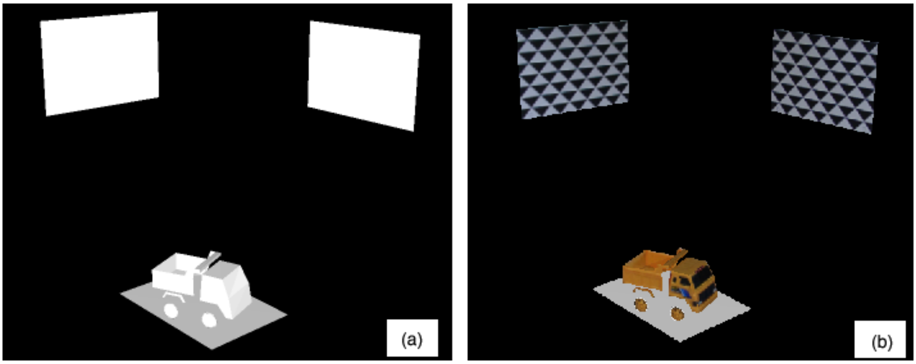


Fig. 4. (a) The model obtained using the 3D model of the **NBM** (b) Texture mapped Model

Table 1. Mean and standard deviation of the differences between the results obtained using the NBM and BBA

	ΔX =BBA$_X$-NBM$_X$	ΔY =BBA$_Y$-NBM$_Y$	ΔZ =BBA$_Z$-NBM$_Z$
$\mu_{BBA-NBM}$	0.013	0.015	0.062
$\sigma_{BBA-NBM}$	0.012	0.013	0.064

Table 2. MSE and Pearson correlation coefficient values between the results obtained using the NBM and BBA

	MSE	Corr
$\Delta X_{BBA-NBM}$	0.000	0.985
$\Delta Y_{BBA-NBM}$	0.003	0.991
$\Delta Z_{BBA-NBM}$	0.003	0.938

image coordinates of the control points have been extracted by employing the well-known least square matching algorithm. On the other hand, the required corrections have been made to the coordinates obtained as a result of the image matching before using the BBA method. The results obtained using the **NBM** and BBA methods have been compared statistically with each other.

As a result, 89.34% of the Δx , 92.84% of the Δy and 81.15% of the Δz values obtained with the **NBM** were found between ($\mu\pm$ 2σ) ,where Δx, Δy and Δz denote the differences between **NBM** and BBA.

Positions of the used 24 control points and the 3D model of the camera positions are illustrated in Figures 2 and 3. In addition, the 3D model of the **NBM** and the texture mapped model are illustrated in Figure 4. Extensive simulations show that the **NBM** supplies statistically acceptable and accurate results as seen in Tables 1-2.

3.1 One Way Multivariate Analysis of Variance (Manova)

In the Manova [24], mean vectors of a number of multidimensional groups are compared. Therefore, the Manova is employed to find out whether the differences in the results of NBM and BBA are statistically significant or not.

The tested null hypothesis is;

H_0: $\mu_1 = \mu_2 = \dots = \mu$

H_1: at least two μ's are unequal

Where μ_1 is the mean vector of the group # 1, μ_2 is the mean vector of the group # 2 and μ is population mean vector.

As a result of the implemented Manova, no statistically significant difference has been found between the results of NBM and BBA. That means that the null hypothesis cannot be rejected. This hypothesis test has been made using Wilk's Λ and χ^2 tests. The details of these tests and Manova can be found in [24].

For the Wilk's Λ test, the test and critical values are computed as 0.97 and 0.94, respectively, given that α significance level is 0.05 and cross-products are 1 and 123. Due to the condition of $\Lambda_{test} > \Lambda_{critic}$, the null hypothesis cannot be rejected.

For the χ^2 test [24], the test and the critical values are computed as 3.40 and 7.81, respectively, given that α significance level is 0.05 and degrees of freedom is 3. Due to the condition of $\chi^2_{test} < \chi^2_{Critic}$, the null hypothesis cannot be rejected. That is to say that there is no statistically significant difference between the results of NBM and BBA. This outcome statistically verifies the advantages of the NBM method in various perspectives.

4 Conclusions

An ANN based camera calibration method for 3D information recovery from 2D images is proposed in this paper. The obtained results have been compared with the traditional BBA. The main advantages of the **NBM** are as follows: It does not require the knowledge of complex mathematical models and an initial estimation of camera calibration, it can be used with various cameras by producing correct outputs, and it can be used in dynamical systems to recognize the position of the camera after training the ANN structure. Therefore, the **NBM** is more flexible and straightforward.

The advantages of the **NBM** may be summarized as follows:

- Does not require expert knowledge
- Suitable for multi-view geometry
- Offers high accuracy
- Simple to apply
- Its accuracy depends on the distribution and number of the control points in addition to the structure of the neural network.
- Suitable for computer vision and small scale desktop photogrammetry
- Does not use traditional calibration methods and parameters.

Acknowledgments

This study has been supported by the Research Foundation of Erciyes University under the project codes of FBT-05-31 and FBA-05-25.

Bibliography

[1] Klir, G., Wang, Z., Harmanec, D.: Geometric Camera Calibration Using Circular Control Points, IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 1066-1076.

[2] Juyang, W., Cohen, P., Herniou, M.: Camera Calibration with Distortion Models and Accuracy Evaluation, IEEE Transactions on Pattern Analysis and Machine Intelligence **14** 10 (1992) 965-990.

[3] Wen, J., Schweitzer, G.: Hybrid calibration of CCD cameras using artificial neural nets, Int Joint Conf. Neural Networks **1** (1991) 337-342.

[4] Lynch, M., Dagli, C.: Backpropagation neural network for stereoscopic vision calibration, Proc. SPIE. Int. Soc. Opt. Eng. **1615** (1991) 289-298.

[5] Ekenberg, L.: Direct Linear Transformation Into Object Space Coordinates in Close-Range Photogrammetry, Proc. Symp. Close-Range Photogrammetry (1971) 1-18.

[6] Ji, Q., Zhang, Y.: Camera Calibration with Genetic Algorithms, IEEE Transactions on Systems, Man, Cybernetics-Part-A: Systems and Humans **31** (2001) 120-130.

[7] Wang, F. Y.: An Efficient Coordinate Frame Calibration Method for 3-D Measurement by Multiple Camera Systems, IEEE Transactions on Systems, Man, Cybernetics-Part-C: Applications and Reviews **35** (2005) 453-464.

[8] Heikkilä, J.: Geometric Camera Calibration Using Circular Control Points, IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 1066-1076.

[9] Zhang, Z.: A Flexible New Technique for Camera Calibration, IEEE Trans. Pattern Analysis and Machine Intelligence **22**, (2000) 1330-1334.

[10] Haralick, R.M.: Pose Estimation From Corresponding Point Data, IEEE Trans. Systems, Man, and Cybernetics **19** (1989) 1425-1446.

[11] Aubin, J.P., Frankowska, H.: Hybrid calibration of CCD cameras using artificial neural nets, Proc. Int Joint Conf. Neural Networks (New York) (1991) 337-342.

[12] Shih, S., Hung, Y., Lin, W.: Accuracy analysis on the estimation of camera parameters for active vision systems, Proc. Int. Conf. Pattern Recognition, Vienna, Austria **1** (1996) 1-25.

[13] Faugeras, O.D., Luong, Q.T., Maybank, S.J.: Camera selfcalibration: Theory and experiments, in Proc. ECCV 92, Lecture Notes in Computer Science **588** (1992) 321-334.

[14] Ahmed, M.T., Farag, A.: Neurocalibration: a neural network that can tell camera calibration parameters, The Proceedings of the Seventh IEEE International Conference on Computer Vision **1** (1999) 463-468.

[15] Heping, P., Chusen, Z.: System Structure and Calibration Models of Intelligent Photogrammetron, Wuhan University Journal **2** (2003).

[16] Pan, H.P.: A Basic Theory of Photogrammetron, International Archives of Photogrammetry and Remote Sensing **34** (2002).

[17] Pan, H.P., Zhang, C.S.: System Calibration of Intelligent Photogrammetron, International Archives of Photogrammetry and Remote Sensing **34** (2002).

[18] Lynch, M.B., Dagli, C.H., Vallenki, M.: The use of fedforward neural networks for machine vision calibration, Int. Journal of Production Economics **60-61** (1999) 479-489.

[19] Fausett, L.V.: Fundamental of neural networks: Architectures, Algorithms, and Applications, Prentice Hall (1994).

[20] Haykin, S.: Neural Networks, a Comprehensive Foundation, Prentice Hall Inc., (1999).

[21] Mathworks Inc., Matlab Neural Networks Toolbox, Matworks, (2005).

[22] Reidmiller, M.: Rprop- Description and implementation details, Technical Report, University of Karlsruhe, Germany, (1994).

[23] Reidmiller, M., Braun, H.: A direct adaptive method for faster backpropogation learning: The Rprop algorithm, Proceedings of the IEEE Int. Conf. On Neural Networks, San Francisco, CA, (1993), 586-591.

[24] Rencher, A.C.: Methods for multivariate analysis, Wiley-Interscience, John Wiley-Sons , Inc., (2002).

Implicit Camera Calibration Using an Artificial Neural Network

Dong-Min Woo and Dong-Chul Park

Dept. of Information Engineering, Myong Ji University, Korea
{dmwoo, parkd}@mju.ac.kr

Abstract. A camera calibration method based on a nonlinear modeling function of an artificial neural network (ANN) is proposed in this paper. With the application of the nonlinear mapping feature of an ANN, the proposed method successfully finds the relationship between image coordinates without explicitly calculating all the camera parameters, including position, orientation, focal length, and lens distortion. Experiments on the estimation of 2-D coordinates of image world given 3-D space coordinates are performed. In comparison with Tsai's two stage method, the proposed method reduced modeling errors by 11.45% on average.

1 Introduction

Camera calibration can be considered as a preliminary step toward computer vision, which makes a relation between real world coordinates and image coordinates. Generally, there are two different kinds of calibration methods: explicit and implicit approaches. Physical parameters of a camera including the image center, the focal length, the position, and the orientation can be obtained through explicit camera calibration [1,2,3,4]. However, physical parameters of a camera are not necessarily available in some stereo vision cases. In this case, we have to use an implicit calibration method. In particular, when the lens distortion is excessive and the image center is assumed to be the center of the frame grabber, it may be difficult to align both the CCD cells and lens in a perfectly parallel position. Some intermediate parameters should be calibrated by estimating image coordinates from known world coordinates. Martins first proposed the two-plane method [5]. Martins' two-plane method considers lens distortions. However, in general, calibrated parameters are not globally valid in the whole image plane. The more recent work of Mohr and Morin [6] can be used for both 3-D reconstruction and the computation of image coordinates. However, lens distortion has not been considered in the implicit camera calibration approach.

In this paper, a new camera calibration approach based on an artificial neural network(ANN) model is proposed. ANNs have been shown to have the ability to model an unspecified nonlinear relationship between input patterns and output patterns. This nonlinear mapping ability can be utilized to address some physical parameters in implicit camera calibration that cannot be readily estimated by the existing calibration methods. The ANN-based camera calibration approach does not estimate camera physical parameters. However, this is not an issue when the

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 641–650, 2006.

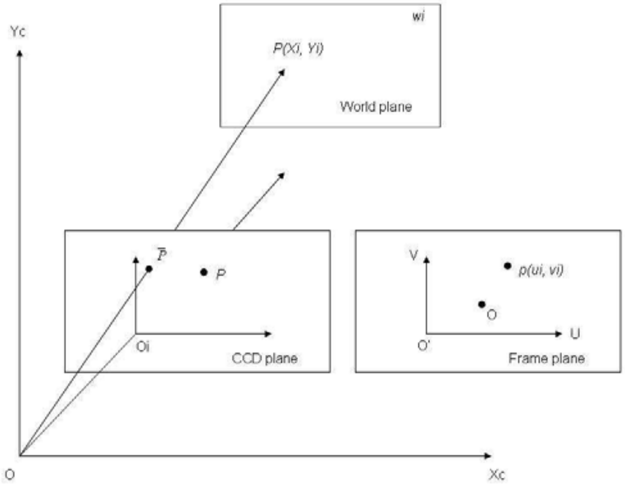


Fig. 1. The imaging geometry

objective of the camera calibration process is to obtain the correlation between the camera image coordinates and the 3-D real world coordinates. The implicit camera calibration approach, which can calibrate a camera without explicitly computing its physical parameters, can be used for both the 3-D measurement and the generation of image coordinates.

The remainder of this paper is organized as follows: Section 2 describes the implicit and explicit calibration methods for the camera calibration problem. Section 3 briefly describes the ANN and the training algorithm adopted in this paper. Section 4 presents experiments and results including the experimental environment used in this work and a performance comparison between the proposed method and Tsai's two stage method. Concluding remarks are given in Section 5.

2 Camera Calibration-Method Using Neural Network

2.1 Implicit Camera Calibration

Suppose that there is a calibration plane and the center of the calibration plane is defined as O. In the calibration plane, we have N points. A point, P:$(X_i, Y_i) \in w_i$, $i = 1, 2, \cdots, N$, in the world plane is ideally projected to $\bar{p} : (\bar{x}_i, \bar{y}_i)$ in the camera CCD plane. However, because of the distance of the camera lens, the point of the world plane is projected to a distorted point, p:(x_i, y_i). This point is observed through the frame buffer coordinate $p(u_i, v_i)$ in pixels, as shown in Fig. 1.

For a back-projection problem, a transformation from the image coordinates in the frame buffer to the world coordinates in the calibration plane is required. For this purpose, an ANN is adopted in the proposed ANN-based calibration

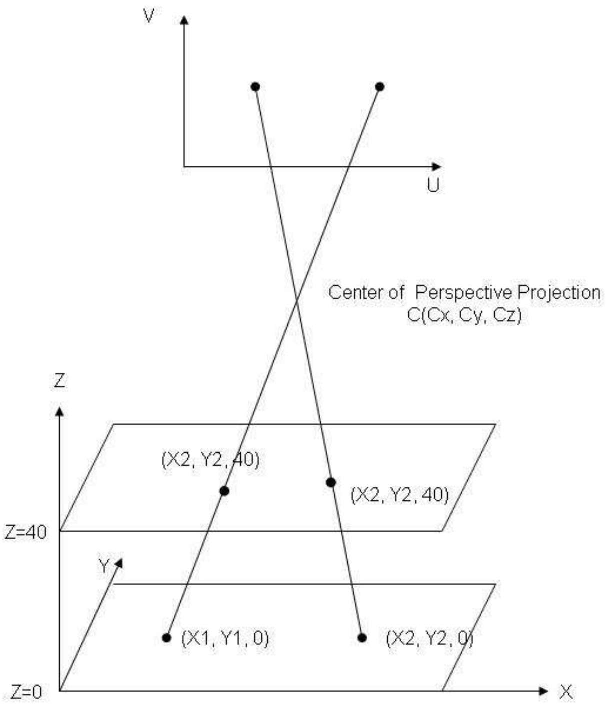


Fig. 2. The center of a perspective projection

approach, where the input and the output of the ANN are the image coordinates and the world coordinates, respectively. After proper training of the ANN with training points, the ANN can map the relation of two planes. Owing to the nonlinear system modeling capability of the ANN, it is not necessary to utilize all the physical parameters involved with the camera calibration, including the lens distortion and the focal length of the camera.

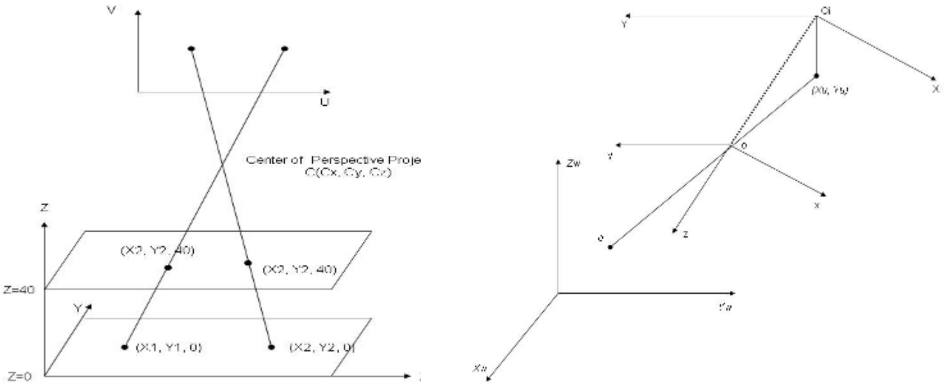
With the coordinate system shown in Fig. 2, $(x_1, y_1, 0)$ and $(x_2, y_2, 0)$ are defined as two points on the calibration plane $Z = 0$ and $(x'_1, y'_1, 40)$ and $(x'_2, y'_2, 40)$ are two other points on the plane $Z = 40$. The line equations that pass each of the two points can be expressed by the following equations:

$$\overrightarrow{\boldsymbol{P}} = (x_{1,}y_{1,}z_1) + t(x'_1 - x_1, y'_1 - y_1, 40) \tag{1}$$

$$\overrightarrow{\boldsymbol{Q}} = (x_{2,}y_{2,}z_2) + t(x'_2 - x_2, y'_2 - y_2, 40) \tag{2}$$

$$\overrightarrow{\boldsymbol{P}} = \overrightarrow{\boldsymbol{Q}} \tag{3}$$

Since the equations given by Eq.(1) and Eq.(2) meet at the point C, i.e., Eq.(3), this point can be considered as the perspective center of the image, as shown in Fig 2.



(a) (b)

Fig. 3. (a) Image coordinate prediction (b) The camera model used in Tsai's two stage model

By using the perspective center of an image, the estimation of the image coordinates of any 3-D world point P can be obtained. In this case, an ANN that is trained with the real world coordinates of points on Z=0 as inputs and the image plane coordinates for the corresponding points as targets is given. It should be noted that the input and target for the ANN in this case are different from those of the back-projection problem. When the image coordinate of a point (P_1) on any calibration plane Z is needed, the line equation that passes the point(P_1) in the calibration plane Z and the perspective center of a camera(C) is first obtained. The line equation can produce P_0 on the calibration plane Z = 0. By using P_0 as the input to the trained ANN, we can obtain the image coordinates of the point $\hat{p}$. This process is shown in Fig. 3-(a).

2.2 Explicit Calibration Method

Tsai's two stage method (TSM)[4], which is considered one of the most powerful methods for explicit camera calibration, is chosen for the purpose of performance comparison. The TSM first obtains the transformation parameters with the assumption that there exists no distortion in the camera. The TSM then refines the transformation parameters with the distortion of the camera by using a non-linear search. That is, first, the camera model is assumed to be ideal for the camera calibration by neglecting the lens distortion.

Fig. 3-(b) shows the camera model used in Tsai's two stage model. In Fig. 3-(b), a point P is an object of the real world coordinate(X_w, Y_w, Z_w) and (x,y,z) is a 3-D camera coordinate. The center of the camera coordinate is the optical center O and (X,Y) is the image coordinate with the center of O_i. The distance between O and O_i is f, the focal length of the camera. (X_u, Y_u) is the corresponding point with the assumption of no lens distortion. (X_u, Y_u) is then translated to (X_f, Y_f), which is a point in computer image coordinate on the image buffer and is expressed in pixel numbers. The basic geometry of the camera model can be written as the transformation of the two coordinates with the following displacement and orientation:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{4}$$

with

$$\begin{aligned} r_1 &= \cos\psi\cos\theta \\ r_2 &= \sin\psi\cos\theta \\ r_3 &= -\sin\theta \\ r_4 &= -\sin\psi\cos\theta + \cos\psi\sin\theta\cos\pi \\ r_5 &= \cos\psi\cos\theta + \sin\psi\sin\theta\sin\pi \\ r_6 &= \cos\theta\sin\pi \\ r_7 &= \sin\psi\sin\pi + \cos\psi\sin\theta\cos\pi \\ r_8 &= \cos\theta\cos\pi \end{aligned}$$

where θ, π, and ψ represent yaw, pitch, and tilta, respectively.

As can be seen from the above equations, there are six extrinsic parameters: θ, π, and ψ for rotation, and three components for the translation vector T. The problem of camera calibration is to find the six parameters θ, π, ψ,T_x, T_y, and T_z by using the number of points measured in the (X_w, Y_w, Z_w) coordinate.

In the second stage of the TSM, a distortion parameter is considered. The relations between the computer image coordinate with distortion and the real world coordinate can be derived as follows:

$$S_x(X_f - C_x)(1 + G(X_d^2 + Y_d^2)) = f\left(\frac{r_1x_w + r_2y_w + r_3z_x + T_x}{r_7x_w + r_8y_w + r_9z_w + T_x}\right) \tag{5}$$

$$S_x(X_f - C_y)(1 + G(X_d^2 + Y_d^2)) = f\left(\frac{r_4x_w + r_5y_w + r_6z_x + T_x}{r_7x_w + r_8y_w + r_9z_w + T_x}\right) \tag{6}$$

where (X_f, Y_f) is the image coordinate of the frame grabber, (C_x, C_y) is the image center, S_x and S_y are components of the translating scale of the x-axis and y-axis when the transform A/D,(X_d, Y_d) is a distorted coordinate by lens distortion, and G is the distortion parameter. Tsai obtained the solution by using a gradient-based nonlinear search method. In an explicit calibration, the calibration is performed with extrinsic parameters. However, the distortion parameters cannot include all the parameters involved in the distortion of the image. Even with the assumption of perfect inclusion of distortion parameters, there still remains room for errors in finding the right solution for such parameters.

3 ANN for Camera Calibration

The ANN model adopted in this paper is a standard MultiLayer Perceptron Type Neural Network (MLPNN) and an error back-propagation algorithm is used for training the MLPNN. After several experiments, the architecture of the MLPNN is selected as 2 × 10 × 8 × 2. Note that the selection of a specific architecture is a state of art and other architectures can be also used without any degradation of the resulting performance. With the architecture chosen, no overfitting problem was experienced with 5,000 training epochs. Note that proper numbers of training epochs are dependent on the complexity of the given problem and the number of training data. Note that the neurons in the input and output layers represent the 2-D coordinates. More detailed information on the MLPNN and error back-propagation algorithm can be found in [7].

Unlike the explicit camera calibration method, the proposed ANN-based method finds the direct relation between the world coordinates and the image coordinates. The ANN adopted in this implicit calibration approach can incorporate all the extrinsic parameters of the camera and the distortion parameters when the ANN is trained properly.

4 Experiments and Results

4.1 System Environment for Experiments

The specifications of the image acquisition tool for our simulation environment are summarized in Table 1.

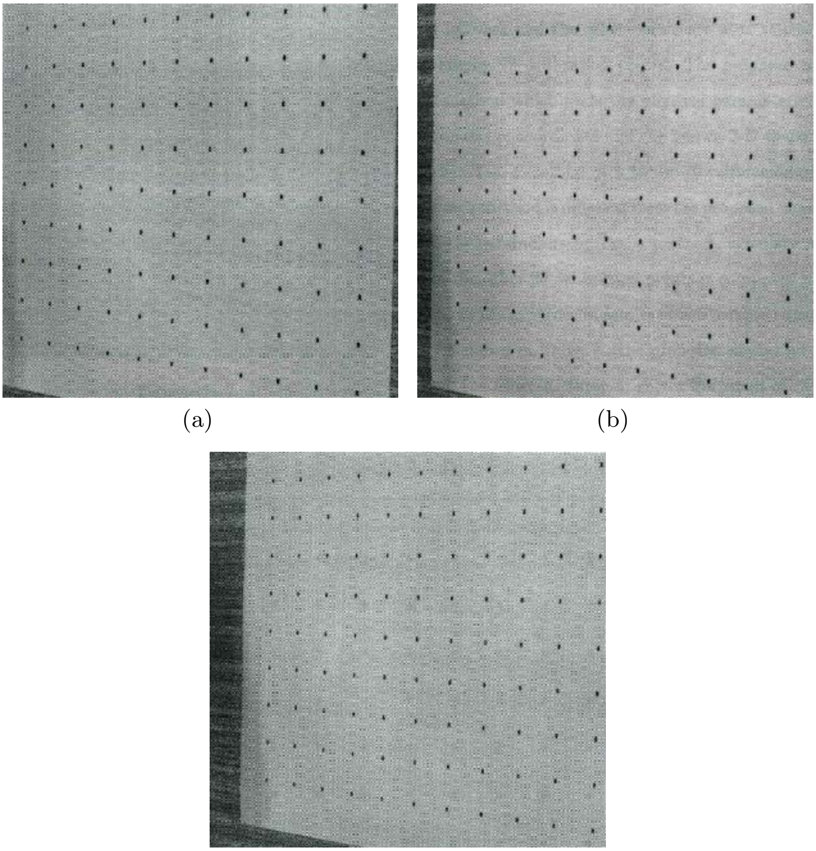
Table 1. The specification of image acquisition

Image aquisition tool	Specification
Frame grabber	Horizontal resolution (X-axis) 512 Vertical resolution 512
CCD Image censor	Scale of cell(X-axis) 17 μm Scale of cell (Y-axis) 13 μm
lens	Focal length (F 1.4) 16mm

Images are acquired at three different positions. The performance of camera calibration results using artificial neural networks is compared and analyzed with that of Tsai's two stage method, the most widely used approach for explicit camera calibration. In this paper, the average error between the calibrated image coordinates and real world coordinates is used to compare the performance of the camera calibration methods. The average error in pixels (AEIP) is defined as follows:

$$AEIP = \frac{1}{N}\sum_{i=1}^{N}[(X_{fi} - \hat{X}_{fi})^2 + (Y_{fi} - \hat{Y}_{fi})^2]^{1/2} \tag{7}$$

where $(\hat{X}_{fi}, \hat{Y}_{fi})$ is the estimated image coordinate, which is computed by using calibrated variables from the real coordinate point (X_{wi}, Y_{wi}, Z_{wi}) corresponding to the computer image coordinate (X_{fi}, Y_{fi}).



(a)

(b)

(c)

Fig. 4. The calibration points at different heights: (a) Z = 40, (b) Z = 20, and (c) Z = 0

The images used for the experiments are obtained by positioning the camera in the real world coordinate. The positions of the camera are also changed along the Z-axis for obtaining image data with different heights. Each image is composed of 99 calibration points (11×9), which have an interval of 25mm between columns and an interval of 20mm between rows. Among the calibration points acquired from two images including 99 calibration points for each different heights, 79 randomly selected calibration points in each image are used for training the ANN and the remaining 20 points are used for evaluation of the trained ANN. Fig. 4 shows the images with different heights used in our experiments.

The proposed method is compared with Tsai's two stages method, which finds the physical parameters of the camera using the interrelation between the image coordinates and the known 3-D space coordinates. For the calculation of the physical parameters for Tsai's method and training ANN, 10 sets of 79 randomly chosen calibration points are collected. For each set of calibration

Table 2. Comparison of estimation errors in AEIP

Case	Tsai's two stage method	ANN-based method
Case #1	0.5322	0.4829
Case #2	0.5277	0.4936
Case #3	0.5576	0.4642
Case #4	0.6201	0.5014
Case #5	0.5970	0.5235
Case #6	0.5128	0.4726
Case #7	0.6026	0.5198
Case #8	0.5844	0.4976
Case #9	0.6214	0.5527
Case #10	0.5993	0.5878
Average	0.5755	0.5096

points, the remaining 20 points are used for testing the performance of both methods. Table 2 shows the test results for both methods. As shown in Table 2, the average improvement of the proposed ANN-based method over Tsai's method in terms of AEIP is 11.45 %.

4.2 Experiments on 3-D Real World Coordinate Reconstruction

The real space coordinate obtained by estimating the 3-D space coordinate at an arbitrary height can be reconstructed after training the ANN with points on two calibration plans, i.e., Z = 0 and Z = 40, as follows: select a certain point of the image and then find the point of the real space coordinate of Z=0 and Z=40 calibration plane corresponding to the selected image point. Using Eq. (1) - Eq. (3), the perspective center of the image can be found. In our experiments, the coordinate of the perspective center is found as $C_x = 556.1233$, $C_y = 53.0954$, $C_z = 634.2666$. When the ANN is trained, ten points of an image are randomly

Table 3. 3-D world coordinate reconstruction error

Real world coordinate Z=20	Result of using ANN	Error
(75,0)	(74.8411, 0.6174)	0.6375
(175,0)	(174.3714, 0.3128)	0.7021
(150,60)	(150.3574, 59.8253)	0.3978
(100,80)	(99.1724, 81.0134)	1.3084
(175,100)	(174.9242, 99.4123)	0.5926
(255,100)	(224.9027, 100.0124)	0.0981
(25,120)	(24.6350, 120.3124)	0.4804
(150,140)	(149.9113, 139.0824)	0.9219
(125,160)	(126.0524, 160.3122)	1.0975
(175,160)	(175.2814, 159.8412)	0.3231
Average error	0.6559	

selected first. The real points for the space coordinate of the $Z = 0$ and $Z = 40$ calibration plane corresponding to the selected image points are then found. By using Eq. (1) - Eq. (3), ten linear equations connecting points of $Z = 0$ plane with points of the $Z = 40$ plane are formulated for estimating the coordinate of the 3-D space on the $Z = 20$ calibration plane. Table 3 shows the estimation results for 3-D space.

4.3 Result and Error of Image Coordinate Estimation

In order to estimate the 2-D image coordinate, the center of perspective projection is first obtained. Ten arbitrary points on the $Z = 20$ plane are selected and the linear equations that connect the selected points with the obtained perspective center in the previous experiment are then found. Finally, intersection points on the $Z = 0$ and $Z = 40$ plane are obtained. Therefore, by using the trained ANN, the image coordinates from the intersecting points of the $Z = 0$ and $Z = 40$ plane are estimated. These experiments are performed with 10 randomly chosen training/test data sets. Table 4 shows the average results over 10 different sets of training/test sets on the 10 data points for coordinates of the 2-D image from the intersecting points of the $Z = 0$ calibration plane and the $Z = 40$ calibration plane.

Table 4. Summary of estimation errors with real world coordinates at z=20

2-D coordinates of image world	Average AEIP
at Z = 0	0.7120
at Z = 40	0.6826

5 Conclusion

A camera calibration method using an artificial neural network is proposed in this paper. The proposed method calibrates a camera using the trained ANN instead of computing the physical camera parameters. The proposed ANN-based implicit method is applied to the estimation of 2-D coordinates of an image world with given 3-D space coordinates. The results are compared with Tsai's widely used two stage method. Results show that the proposed method can reduce the modeling errors by 11.45 % on average in terms of AEIP. The proposed method has advantages over Tsai's two stage method in real-time applications as it can be operated in real time after proper training while Tsai's two stage method requires somewhat time consuming procedures for calculating proper parameters for a given task. The proposed method is also more flexible than Tsai's two stage method since it is not affected by camera position, illumination or distortion of the camera lens. More importantly, the proposed ANN-based method is not affected by the quality of the camera lens in finding the mapping function between the image coordinates and the real coordinates whereas Tsai's method is considerably affected by the quality of the camera. In comparison to the conventional approach Tsai's two stage method, the proposed ANN-based method shows promising results for calibrating camera when issues including practical applicability, flexibility, and real-time operation are relevant.

Acknowledgment

This work was supported by the Korea Research Foundation Grant funded by the Korea Government (MOEHRD-KRF-2005-042-D00265).

References

1. Faig, W.: Calibration of Close-Range Photogrammetry Systems: Mathematical Formulation. Photogrammetric Eng. Remote Sensing, Vol. 41 (1975) 1479-1486
2. Sobel,I.: On Calibrating Computer Controlled Cameras for Perceiving 3-D Scenes. Artificial Intell., Vol. 5 (1974) 185-188
3. Itoh, J., Miyachi, A., Ozawa, S.: Direct Measuring Method using Only Simple Vision Constructed for Moving Robots. Proc. 7th Int. Conf. On Pattern Recognition, Vol. 1 (1984) 192-193
4. Tsai, R.: An Efficient and Accurate Camera Calibration Technique for 3-D Machine Vision. Proc. IEEE Int. Computer Vision and Pattern Recognition, (1986) 364-374.
5. Martins, H.A., Birk, J.R., Kelley, R.B.: Camera Models Based on Data from Two Calibration Plane. Computer Graphics and Image Processing, Vol. 17 (1981) 173-180
6. Mohr, R., Morin, L.: Relative Positioning from Geometric Invariant. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (1991) 139-144
7. Rumelhart, D., Hinton, G., Williams, R.: Parallel Distributed Processing. Cambridge, MIT Press (1986)

3D Freeform Surfaces from Planar Sketches Using Neural Networks

Usman Khan, Abdelaziz Terchi, Sungwoo Lim, David Wright, and Sheng-Feng Qin

Brunel University, Uxbridge, Middlesex, United Kingdom
{Usman.Khan, Aziz.Terchi, Sungwoo.Lim, David.Wright,
Sheng.Feng.Qin}@brunel.ac.uk

Abstract. A novel intelligent approach into 3D freeform surface reconstruction from planar sketches is proposed. A multilayer perceptron (MLP) neural network is employed to induce 3D freeform surfaces from planar freehand curves. Planar curves were used to represent the boundaries of a freeform surface patch. The curves were varied iteratively and sampled to produce training data to train and test the neural network. The obtained results demonstrate that the network successfully learned the inverse-projection map and correctly inferred the respective surfaces from fresh curves.

Keywords: neural networks, freeform surfaces, sketch-based interfaces.

1 Introduction

The preliminary stages of the conceptual product design process are characterised by a high degree of creative activity. Designers strive to convert new ideas into graphical form as soon as possible. It can be argued that sketching is an essential activity for creative design. The reasons are manifold. It permits the rapid exploration and evaluation of concepts [1]. It also assists the designer's short-term memory and facilitates communication with other people. When designers sketch shapes on a sheet of paper, they start with a vague concept, which they progressively refine into a final product. While numerous iterations are usually undertaken, the salient properties of the original idea are often maintained. Recently, the desire to automate the early phase of the conceptual product design have given impetus to the development intelligent tools to simulated the way of sketching is performed by designers [2-4]. However, most existing approaches are restricted to fairly simple objects such as planar and polygonal shapes. Consideration of complex free-form surfaces is a challenging process. The problem has surprisingly received little attention in the literature.

The problem of reconstructing a three dimensional (3D) shape from a planar drawing is fundamental problem in computer vision and computer aided geometric design. Clowes [5], developed a classification method based on labelling drawings and sorting their edges to recognise polyhedral shapes. Though, their method was extended to other line drawings [6-8], their work mainly involved determination of the depth from a 2D drawing consisting of flat surfaces with straight line edges. With regard to freeform surfaces some of the foundation work was developed by Igarashi *et al.* [3] who reproduced rough freeform models from freehand sketch input. Since then

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 651 – 660, 2006.

only moderate progress has been achieved in recovering freeform surfaces from online sketches. Michalik *et al.* [9] proposed a constraint-based system that reconstructed a B-spline surface from a sketch into 3D. These papers employ techniques based on rules or constraints to extract the correlation between the 2D drawings and their respective 3D shapes. In the same vein, the work of Lipson and Shpitalni [6] is also based on the notion of correlation.

Work in recognition of shape features from 2D input was reported by Nezis and Volniakos [10]. The topology of the input drawing was exploited to categorize the shape features. Peng and Shamsuddin [11] claimed that a neural network was able to estimate the pose of a 3D object from a 2D image from any arbitrary viewpoint. Reconstruction of 3D shapes by estimating their depth was done by Yuan and Niemann [12]. They represented objects using a triangular mesh from reverse engineered data and demonstrated that a neural network could reconstruct 3D geometry from 2D input.

Early work pertaining to reconstruction of freeform surfaces was covered by Gu and Yan [13]. A non-uniform b-spline (NURB) surface was fitted over scattered data from a reverse engineering source using an unsupervised neural network. Hoffman and Varady [14] and Barhak and Fischer [15] extended this line of research. However, their methods required that all three dimensions be available for reconstruction purposes.

The present paper proposes and develops a methodology for 3D freeform surface inference from freehand planar sketches. The methodology is based on neural networks. Specifically, an MLP neural network, trained with a momentum-augmented backpropagation learning algorithm, is employed to induce 3D freeform surfaces from 2D sketches. The reconstruction procedure consists of two steps: first a neural network is trained on pairs of normalised 3D surfaces and their corresponding projection curves, then the trained neural network is used to reconstruct unknown 2D sketches. The methodology is tested with a range of data and produced satisfactory results.

The remainder of this paper is organised as follows. In section 2 3D freeform surface reconstruction is formulated as an inverse problem. In section 3, neural networks together with their learning algorithms are discussed. The data generation procedure is discussed in section 4. The computational results are presented in section 5. Finally section 6 treats conclusions and future work.

2 Problem Formulation

Volumetric concepts originate in the mind of a designer as 3D entities. They are then transformed, via an isometric projection onto an arbitrary view plane, into planar sketches. Such a task is considered as the direct problem. The 3D freeform surface inference problem consists of extracting the 3D geometry from the 3D, i.e., to recover the depth information that was lost during the projection process. This process can be regarded as the inverse process of the original projection. The direct problem is, in general, a well-posed problem and can be solved analytically using concepts from projective geometry.

In contrast, the inverse problem is, in general, ill-posed. The solution may not be unique, may lack continuity could be highly influenced by the amount of noise present in the data. Therefore, 3D surface reconstruction is indeterminate in that an infinite number of possible 3D surfaces can correspond to the same 2D curve. To obtain a unique and physically meaningful solution requires additional information in terms of general assumptions, constraints and clues from experience. In the context of this paper, the planar curves are constrained to lie in the x-z or the y-z planes and their control points are restricted to vary only along the z-direction. Such constraint ensures the maintenance of the planar property of the inferred 3D surfaces and leads to a single one to one mapping from the input 2D curves to the expected 3D surfaces. This renders the inverse problem tractable.

Given a set of p ordered pairs $\{(\mathbf{x}_i, \mathbf{y}_i), i = 1,\ldots, p\}$ with $\mathbf{x}_i \in \mathrm{R}^2$ and $\mathbf{y}_i \in \mathrm{R}^3$, the surface reconstruction problem is to find a mapping $F : \mathrm{R}^2 \rightarrow \mathrm{R}^3$ such that $F(\mathbf{x}_i) = \mathbf{y}_i$, $i = 1,\ldots, p$. In practice, the function F is unknown and must be determined from the given data $\{(\mathbf{x}_i, \mathbf{y}_i), i = 1,\ldots, p\}$. A neural network solution of this problem is a two-step process: training, where the neural network learns the function from the training data $\{\mathbf{x}_i, \mathbf{y}_i\}$, and generalisation, where the neural network predicts the output for a test input. We demonstrate how an MLP neural network trained with a momentum-augmented backpropagation algorithm on a collection of 2D-3D dependencies, can approximate the inverse map in a computationally efficient form.

3 Neural Networks

Neural networks are connectionist computational models motivated by the need to understand how the human brain might function. A neural network consists of a large number of simple processing elements called neurons. Feedforward neural networks have established universal approximation capability [16] and have proven to be potent tool in the solution of approximation, regression, classification and inverse problems.

For this reason, a MLP neural network is selected for the solution of the reconstruction problem. The MLP neural network is composed of three layers: the input layer, the hidden layer and the output layer. The neurons of the input layer feed data to the hidden layer where it performs the following nonlinear transformation:

$$s_j = f\left(\sum\nolimits_k w_{jk} x_k\right) . \tag{1}$$

where $\mathbf{x}_k$ are the neurons inputs signals, $\mathbf{s}_j$ are neural outputs and $\mathbf{w}_{jk}$ the synapses and f is an activation function. For MLP neural network, the sigmoid function is used as the activation function. The output layer of the neurons takes the linear transformation:

$$y_j = f\left(\sum\nolimits_k w_{jk} s_k\right) . \tag{2}$$

where $\mathbf{y}_\mathrm{j}$ are the output layer neuron outputs, and $\mathbf{w}_\mathrm{ij}$ are synapses. Neural network training can be formulated as a nonlinear unconstrained optimisation problem. So the training process can be realised by minimising the error function E defined by:

$$E = \frac{1}{2}\sum_{k=1}^{p}\sum_{j=1}^{n}\left(y_{jk} - t_{jk}\right)^2 . \tag{3}$$

where $\mathbf{y}_{jk}$ is the actual output value at the j-th neuron of output layer for the k-th pattern and $\mathbf{t}_{jk}$ is the target output value. The training process can be thought of as a search for the optimal set of synaptic weights in a manner that the errors of the output is minimised.

3.1 Backpropagation Algorithm

Most learning algorithms are based on the gradient descent strategy. The backpropagation algorithm (BP) [17] is no exception. The BP algorithm uses the steepest descent search direction with a fixed step size α to minimise the error function. The iterative form of this algorithm can be expressed as:

$$w_{k+1} = w_k - \alpha g_k . \tag{4}$$

where **w** denotes the vector of synaptic weights and $g = \nabla$ E(**w**) is the gradient of the error function E with respect to the weight vector **w**.

In the BP learning algorithm the weight changes are proportional to the gradient of the error. The larger the learning rate, the larger weight changes on each iteration, and the quicker the network learns. However, the size of the learning rate can also influence the network's ability to achieve a stable solution. In a neighbourhood of the error surface where the gradient retains the same sign, a larger value of the learning rate α results in a rapid reduction of the energy function faster. On the other hand, in an area where the gradient rapidly changes sign, a smaller value of α maintains the descent direct along the error surface.

Despite its computational simplicity and popularity, the BP training algorithm is plagued by such problems as slow convergence, oscillation, divergence and "zigzagging" effect. The BP learning algorithm is in essence a gradient descent optimisation strategy of a multidimensional error surface in the weight space. Such strategy exhibits has inherently slow convergence; especially on large-scale problems. This trait becomes more pronounced when the condition number of the Hessian matrix is large. The condition number is the ratio of the largest to the smallest eigenvalue of the network's Hessian matrix. The Hessian matrix is the matrix of second order derivatives of the error function with respect to the weights.

In many cases the error hypersurface is no longer isotropic but rather exhibits substantially different curvatures along different directions, leading to the formation of long narrow valleys. For most points on the surface, the gradient does not point towards the minimum, and successive steps along the gradient descent oscillates from one side to the other. Progress towards the minimum becomes very slow. This suggests a method that dynamically adapts the value of the learning rate, α to the topography of the error surface.

3.2 Momentum-Augmented Backpropagation

One way to circumvent the above problem, the BP propagation in eq. 4 is augmented with *a momentum term*:

$$w_{k+1} = w_k - \alpha g_k + \beta(w_k - w_{k-1}) \,. \tag{5}$$

The momentum term, β has the following effects: 1) it smoothes the oscillations across narrow valleys; 2) it amplifies the learning rate when all the weights change in the same direction; and 3) enables the algorithm to escape from shallow local minima.

In essence, the momentum strategy implements a variable learning rate implicitly. It introduces a kind of 'inertia' in the dynamics of the weight vector. Once the weight vector starts moving in a particular direction in the weight space, it tends to continue moving along the same direction.

If the weight vector acquires sufficient momentum, it bypasses local minima and continues moving downhill. This increases the speed along narrow valleys, and prevents oscillations across them. This effect can also be regarded as a smoothing of the gradient and becomes more pronounced as the momentum term approaches unity. However, a conservative choice of the momentum term should be adopted because of the adverse effect that might emerge: in a narrow valley bend the weight movement might jump over the walls of the valley, if too much momentum has been acquired.

The learning algorithm requires the *a priori* selection of the learning rate and the momentum coefficient. However, it may not easy to choose judicious values for these parameters because a theoretical basis does not seem to exists for the selection of optimal values. One possible strategy is to experiment with different values of these parameters to determine their influence on the overall performance. The moment augmented backpropagation algorithm may be used both in batch and on-line training modes. In this paper the batch version is used.

4 Data Generation

The neural network used in this paper is trained in a supervised mode via a collection of input-output pairs to optimise the network parameters (i.e. synaptic weights and biases). Training is accomplished through a learning algorithm that iteratively adjusts the network parameters until the mean squared error (MSE) between the predicted and the desired outputs reaches a suitable minimum.

A training set was generated from a family of freeform surfaces whose edges also referred to as the boundaries, consisted of four orthogonally arranged planar curves. An example of a planar curve is shown in Fig. 1. Each curve was governed by four independent control points and represented by a Non Uniform Rational B-Spline (NURBS). Two control points determined the ends of the curve whereas the remaining ones controlled its general shape. NURBS control points need not intersect the curve and can lie anywhere in the 3D space. The curve was uniformly sampled and the coordinates of the sample points formed the input features for the neural network.

The planar curves were placed in the x-z plane or the y-z plane and their control points were only altered along the z-direction to maintain their planar property. Each of the four boundary curves were uniformly sampled at 10 positions. Hence a surface, whether represented in 2D or 3D, consisted of 40 sample points. A point on the 3D surface is represented by the x, y and z coordinates whereas in 2D, it is represented by its x and y coordinates. Therefore a 3D surface is represented by 120 independent features and its respective 2D curve by 80 features.

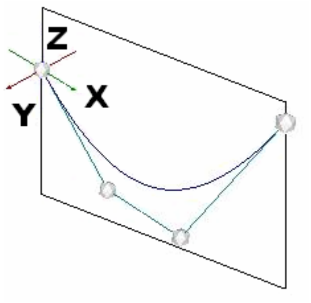
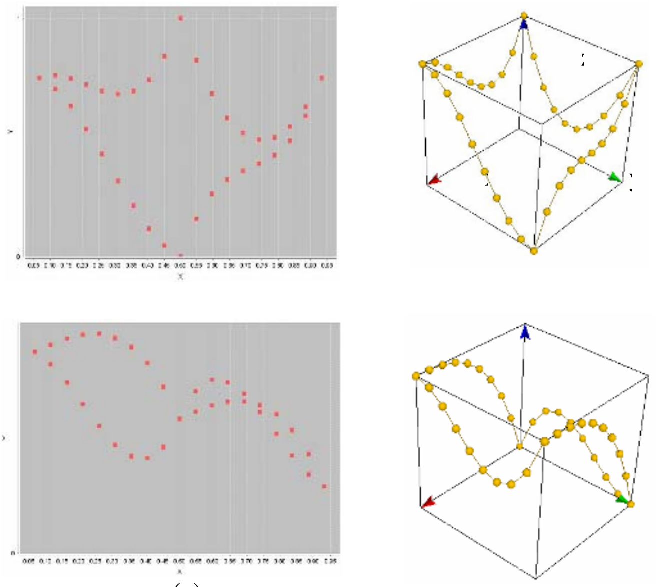


Fig. 1. Planar 3D NURBS curve. Each control point of the curve lies on the same plane as the others.

The positions of the control points were varied to produce a class of unique freeform surfaces. Each surface was projected onto the view plane to produce the respective 2D planar projection. The training set is composed of pattern pairs, each containing a 3D surface and its corresponding 2D curve.

The data set was normalised so that the input 3D pattern would fit within a unit cube and its respective 2D pattern within the unit square. Normalisation ensures that the values lie within the characteristic bounds of the activation functions.

Fig. 2 shows two examples of normalised pattern pairs that were used to train the neural network. The 2D input patterns are depicted in Fig. 2 (a) whereas their



(a) (b)

Fig. 2. Examples of 2D input patterns and corresponding 3D output patterns

corresponding 3D output patterns are shown in Fig. 2 (b). It can be seen that the boundary of the surfaces are described by a series of sample points and fits within a unit square for 2D and unit cube for 3D. Notice that the viewpoint of the 3D desired pattern coincides with the viewpoint of the 2D input pattern.

The entire data set was composed of 4096 patterns. The whole set cannot be used to train the network because no data would be left to test the network's ability to generalise into fresh inputs. Therefore the data set was randomly split, using three subsets that were used for training, validation and testing. Accordingly, the number of training, validation and testing patterns pairs were therefore 2867, 819 and 410 respectively. This corresponds to a 70, 20 and 10 percent split of the data.

5 Computational Results

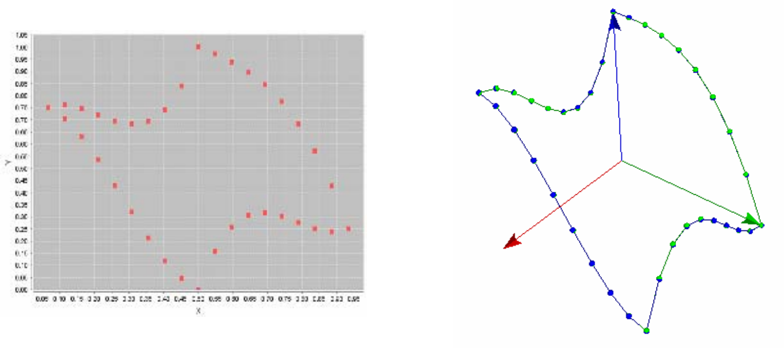
A three-layer MLP network was employed in our research. The input and output layer dimensions of the neural network were determined from the features of the training set. The input layer consist of 80 nodes and while the output layer consists of 120 nodes. The number of nodes in the hidden layer is freely adjustable and results in different network performance depending on the number of hidden nodes used. The parameters used in the network are shown in Table 1.

Table 1. Network Architecture and Parameters

Number of Input Nodes	80
Number of Output Nodes	120
Learning Rate (α)	0.7
Momentum (β)	0.6
Number of Epochs	5000
Number of Training Patterns	2867
Learning Mode	Batch

The number of hidden nodes indicates the network complexity and governs how accurately it learns the mapping from the input patterns to the outputs. It also affects how long the network takes to perform each training cycle. The higher the number of hidden nodes, the more computation is required and hence a longer training time is needed. Experimentation with different numbers of nodes in the hidden layer was conducted. Multiple neural networks were trained with similar parameters such as the learning rate, momentum and training sets. In this case the learning rate was 0.7 and the momentum was 0.6. Only the number of hidden nodes was changed. It was found that a neural network of 50 hidden nodes produced the best reconstruction error over a fixed number of epochs. This was found by comparing the average reconstruction error of the networks based on a fresh test set containing 410 patterns.

Finally a new network of 50 hidden units was trained again for 5000 epochs. The final training error was 0.06. At the end of the training, the net was saved the test set applied to the network. The obtained results show that the neural network was able to infer the 3D shape of a freeform surface from its respective 2D input pattern.



(a)

(b)

Fig. 3. Test Input Patterns with Predicted and Desired Outputs

An example test pattern that was applied to the trained network is shown in Fig. 3 (a). The predicted and the expected 3D patterns that correspond to the 2D surface are shown in Fig. 3 (b). The predicted pattern is depicted in green whereas the desired pattern is in blue. It can be noticed from the plots in Fig. 3 (b) that the two surfaces per image are almost identical and hence that the neural network has inferred the correct shape that was desired. However, small deviations in the predicted patterns can be seen when observed closely. They relate to the network's ability to predict the desired surfaces. The distributions of errors are presented in Fig. 4. This shows the Euclidean distances between each point from the predicted surface and its corresponding point on the desired surface. The RMS error for this pattern was 0.33%.

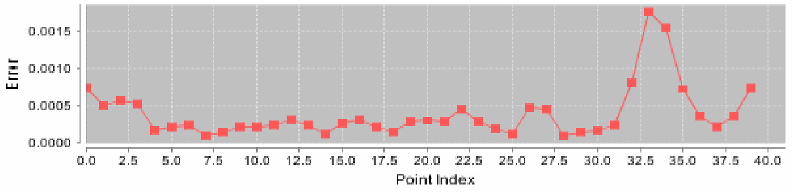


Fig. 4. Distribution of Squared Errors Between Predicted Output and Expected Output

6 Conclusions and Future Work

In this paper a methodology for the inference of 3D freeform surfaces from 2D surface representations using neural networks has been proposed. A representative dataset was generated by iteratively adjusting the control points of freeform surface boundary curves that were previously uniformly sampled. The dataset was normalised and randomly split into three subsets: training, validation and test sets. An MLP was optimised using different numbers of hidden nodes. The best network, i.e. the network with the lowest training RMSE, was trained with a representative family of 2D and 3D pattern pairs. The neural network was applied to a set of 2D patterns had not been encountered before. Obtained 3D results demonstrate that the target freeform surfaces can be reproduced from 2D input patterns within 2 % accuracy. Future work will extend the methodology to more complex shapes and reconstruct the 3D surface that corresponds to the inferred surface boundary.

Acknowledgements

This research is supported by the EPSRC, research grant [EPSRC GR/S01701/01]. The methodology was implemented in Java 1.5 using Netbeans 5.0 based on a Windows 2000/XP platform. The neural network was developed using the Java Object Orientated Neural Engine and the 3D graphics drawn using Jun for Java. The authors would also like to thank Yosh Nishinaka from Software Research Associates Inc. and Paolo Marrone from JOONE for their assistance.

References

1. Lim, S., Lee, B., Duffy, A.: Incremental modelling of ambiguous geometric ideas (I-MAGI): representation and maintenance of vague geometry. Artificial Intelligence in Engineering **15** (2001) 93-108
2. Karpenko, O., Hughes, J., Raskar, R.: Free-form Sketching with variational implicit surfaces. Eurographics **21** (2002) 585-594
3. Igarashi, T., Matsuoka, S., Tanaka, H.: Teddy: A Sketching Interface for 3D Freeform Design. 26th International Conference on Computer Graphics and Interactive Techniques (1999) 409-416
4. Alexe, A., Gaildrat, V., Barth, L.: Interactive Modelling from Sketches using Spherical Implicit Functions. Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa. Proceedings of the 3rd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa., Africa (2004) 25-34
5. Clowes, M.: On Seeing Things. Artificial Intelligence **2** (1971) 79-116
6. Lipson, H., Shpitalni, M.: Correlation-Based Reconstruction of a 3D Object from a Single Freehand Sketch. AAAI Spring Symposium on Sketch Understanding, Palo, Alto, USA (2002) 99-104
7. Varley, P., Martin, R.: Estimating Depth from Line Drawings. Symposium on Solid Modelling. ACM press, Saarbrucken, Germany (2002) 180-191
8. Shpitalni, M., Lipson, H.: 3D conceptual design of sheet metal products by sketching. Journal of Materials Processing Technology **103** (2000) 128-134
9. Michalik, P., Kim, D.H., Bruderlin, B.D.: Sketch- and constraint-based design of B-spline surfaces. Proceedings of the seventh ACM symposium on Solid modelling and applications. ACM Press, Saarbrücken, Germany (2002) 297-304
10. Nezis, K., Vosniakos, G.: Recognizing 2 1/2D shape features using a neural network and heuristics. Computer-Aided Design **29** (1997) 523-539
11. Peng, L.W., Shamsuddin, S.M.: 3D Object Reconstruction and Representation Using Neural Networks. Computer graphics and interactive techniques in Australia and South East Asia, Singapore (2004) 139-147
12. Yuan, C., Niemann, H.: Neural Networks for appearance-based 3-D object recognition. Neurocomputing **51** (2003) 249-264

13. Gu, P., Yan, X.: Neural network approach to the reconstruction of freeform surfaces for reverse engineering. Computer Aided Design **27** (1995)
14. M. Hoffman, Varady, L.: Free-form Surfaces for Scattered Data by Neural Networks. Journal for Geometry and Graphics **2** (1998) 1-6
15. Barhak, J., Fischer, A.: Adaptive reconstruction of freeform objects with 3D SOM neural network grids. Computer and Graphics **26** (2002) 745-751
16. Hornick, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks **2** (1989) 359-366
17. Rumelhart, D.E., McClelland, J.L.: Parallel Distributed Processing: Exploration in the Microstructure of Cognition, Vol. 1. MIT Press, Massachusetts (1986)

General Adaptive Transfer Functions Design for Volume Rendering by Using Neural Networks

Liansheng Wang, Xucan Chen, Sikun Li, and Xun Cai

School of Computer Science,
National University of Defense Technology, 410073 Changsha, China
wangliansheng_nudt@yahoo.com

Abstract. In volume data visualization, the classification is used to determine voxel visibility and is usually carried out by transfer functions that define a mapping between voxel value and color/opacity. The design of transfer functions is a key process in volume visualization applications. However, one transfer function that is suitable for a data set usually dose not suit others, so it is difficult and time-consuming for users to design new proper transfer function when the types of the studied data sets are changed. By introducing neural networks into the transfer function design, a general adaptive transfer function (GATF) is proposed in this paper. Experimental results showed that by using neural networks to guide the transfer function design, the robustness of volume rendering is promoted and the corresponding classification process is optimized.

Keywords: classification; transfer functions; visualization; neural network.

1 Introduction

A volumetric data object is described as a space-filling three-dimensional grid of discrete sample points, which, in turn, support the interpolation of any arbitrary point within the grid's 3D bounding box. A great variety of disciplines generate, use, and modify volumetric data. Examples are the medical field in diagnosis and surgical simulation, engineering in CAD/CAM prototyping, the oil and gas industry in natural resource exploration, designers in virtual sculpting design, the computer game industry in the generation of realistic natural phenomena, computational scientists in scientific data exploration, and the business world in visual data mining.

Direct volume rendering is a key technology for visualizing large 3D data sets from scientific or medical applications, which allows scientists to gain insights into their data sets through the display of materials of varying opacities and colors. Volumetric data sets often have a single scalar value per voxel, so classification of these voxels to assign color and opacity is critical in obtaining useful visualizations that help to provide understanding into a data set. Without a proper classification function to show interesting features or remove obscuring data, it is impossible to correctly interpret the volumetric content.

Transfer functions (TFs) are typically employed to perform this task of classification, which are particularly important to the quality of direct volume-rendered images. A transfer function (TF) assigns optical properties, such as color and

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 661 – 670, 2006.

opacity, to original values of the data set being visualized. Unfortunately, finding good TFs proves difficult. Pat Hanrahan called it one of the top 10 problems in volume visualization in his inspiring keynote address at the 1992 Symposium on Volume Visualization. And it seems that today, almost a decade later, there are still no good solutions at hand.

The most common scheme for TF specification is by trial and error, and other methods are also used to generate TF. But, automatic design of a high performance TF has been proved difficult. First, in many cases, little pre-knowledge makes it difficult to obtain information and gain understanding of the data set. Second, the same data value may belong to different structures or matters, in reverse, the same structure or matter may present the same data value due to noise, so automatic segmentation and classification of arbitrary volume data are still difficult in science. Third, complexity of the volume rendering process results in the nonlinear relationship between the optical properties produced by the transfer function and the final rendering image. Trivial tune of transfer function may lead to tremendous change in the final rendering. And another important fact is that a new TF designed for one type of dataset in most cases can't be used to render other types. It means that you may need to design different TFs for different types of datasets, which consumes much time.

While this paper aims to propose a new TF design method for volume rendering based on neural network, our main contributions are two techniques which use Kohonen's Self-Organizing Map (SOM) network [22] to identify the type information of a data set and use Back Propagation Neural Networks (BPN) to classify the data set and assign optical properties to it. Section 2 surveys current volume TF research progress. Our new method is presented in section 3. In section 4, the experiments and results are presented. Finally the conclusion and future work in section 5.

2 Related Work

Current literature reports various efforts being made toward the construction of optimal TFs.

(1) The most common scheme for TF specification is by trial and error [20]. This involves manually editing a typically linear function by manipulating "control points" and periodically checking the resulting volume rendering. Even if specialized hardware support is available, e.g. a VolumePro board [8], this method can be very laborious and time-consuming. The problem lies in the lack of a precise correspondence between control point manipulation and its effects on the rendered images.

(2) Data centric without data model [18]. This method uses a gradient integral function scheme which automatically discriminates between diverse materials within the data set from which appropriate color and opacity maps are obtained. However, due to the association of isosurfaces with isocontours display, not all the voxels may contribute to the final rendered volume. Bajaj et al. [18] describe a tool for assisting the user in selecting isovalues for effective isosurface volume visualizations of unstructured triangular meshes for isosurface rendering. Fujishiro et al. [10] use a

"Hyper Reed graph" to depict the isosurface topology at any given isovalue, as well as the isovalues corresponding to critical points where the topology changes.

（3） Data Centric with data model [21]. This work expands and generalizes three previously proposed methods: barycentric opacity maps, hue-balls (for color), and lit-tensors (for shading). It is a semiautomatic process that constructs TFs and OP values with an edge detection algorithm. Boundaries, opacity and shading are located in a 1D space of the data set. The method experiences difficulties with data sets that include noise and coarse boundary samples such as certain MRI samples. Kindlmann et a1.[11][12] demonstrate an innovate semi-automatic transfer function design method based on the analysis of a three-dimensional histogram which records the correlation, throughout the given data set, between data value, gradient magnitude, and the second directional derivative along the gradient direction. The method calculates opacity functions which aim to only make those positions in the transfer function domain opaque which reliably correspond to the boundary between two relatively homogeneous regions. Sato et al. [13] use weighting functions of eigenvalues of the Hessian matrix to measure the shape of local structures in terms of edge, sheet, line, and blob. Two or three of these measures can be used as axes of a transfer function emphasize different structures in the volume according to their shapes, which tends to have biological significance in the context of medical imaging. Pekar et al. [14] propose a method which requires only a single pass through the data set to create a Laplacian-weighted histogram of data values to guide the isovalue selection and opacity function creation. Hladiivka et al. [4] use two-dimensional space of principal surface curvature magnitudes (KO, K2) to form a transfer function domain, which is trivial to enhance and color different structures in the volume according to their surface shape. Jiawan Zhang et a1. [15] propose a semi-automatic data-driven transfer function design method by examining relationships among three eigenvalues of inertia matrix. Local features detected by local block based moments, such as flat, round, elongated shapes are used to guide the design of transfer functions.

（4） Image-centric using organized sampling [9]. He et al. [5] describe the search of good transfer functions as a parameter optimization problem. One of common genetic algorithms-stochastic search is used to achieve global optimization. Here, TFs are designed and implemented with the Design Gallery method which contains the volume rendering construct VolDG. With VolDG, (a) TFs may be generated automatically; (b) manipulation of color and opacity TFs generates gray scale images. However, generating time may be significant for complex data sets (7 hours for 20,000 iterations). Fang et a1. [7] present an image-based transfer function model based on three dimensional(3D) image processing operations. Konig and Groller [19] organize the rendered thumbnails efficiently to guide the transfer function process based on volume hardware.

The methods just presented reflect real significant advances in volume rendering. However, they did not answer the question of which one is suitable for an unknown data set. That is, when a new data set is given, which method is the most efficient to reveal the underlined data set essence? How much does each parameter in the transfer function domain contribute to the final rendered image?

In this paper, by introducing Kohonen's SOM and the BPN into the TF design, we obtain preliminary answers to the questions above.

3 A General Adaptive Transfer Function (GATF)

The dataflow of our approach is shown in Fig. 1. In the Machine Learning engine, the Information Bank is designed to store the type information of the data sets and the weights of the BPN for each data set. The module SOM identifies the type of the date set, and the module BPN generates a proper TF for the data set consequently. During rendering process, the adaptive transfer function is exploited to assign opacity to every voxel.

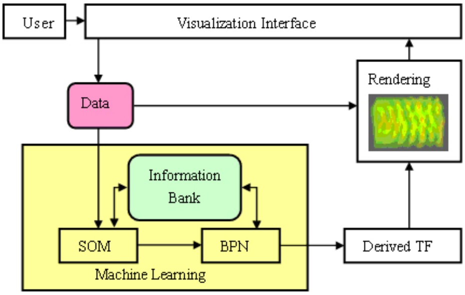


Fig. 1. The visualization process based on our approach. Users first input a data set, and then the machine learning engine generates an adaptive transfer function automatically for volume rendering.

3.1 The Module Som

As the first module of the Machine Learning engine, Kohonen's SOM [22] is responsible for identifying the data set, of which the network is shown in Fig. 2. On receiving a data set, the SOM trains itself to get the data set type. The type information generated by the SOM is then compared with the obtained type information stored in the Information Bank. A similarity value is computed for the data set type and each one in the Information Bank. On one hand, if the maximum similarity value is less than the desired one, it indicates that the data set is of new type, and it is sent to the BPN and its type information is stored into the Information Bank at the same time. On the other hand, if the similarity value is larger than the desired one, the remaining task is to give this information to the BPN module. Hence the key work of this module is how to get the similarity value.

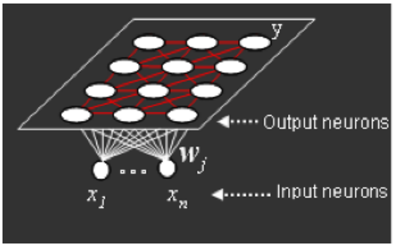


Fig. 2. The structure of SOM network

The competitive learning SOM is proposed by Kohonen [22], which plays an important role as a component in a variety of natural and artificial neural information processing systems. The underlying principle of SOM (and its variants) is the preservation of the probability distribution and topology. The SOM model used in our paper is a simple single layer network in which x_i is the input vector and each neuron has a weight vector W_j, $j=1,\ldots,m$, where m is the number of neurons. For an input data p, we first compute the similarity between p and each neuron j, and then we accumulate all the similarities. Considering Euclidean distance often be a estimated function in the self-organizing training of SOM, we first compute the Euclidean distance between p and neuron j, which is d_j = ||p- w_j ||, $j=1,2,\ldots,m.$. Then we use the Euclidean distance d_j to compute the similarity between p and neuron j, in which $f(d_j)$ is a decreasing function. It means that the shorter distance (d_j) the larger similarity value. In this paper, we choose $f(d_j)$ as a typical sigmoid function

$$f(d_j)=\frac{1}{1+e^{d_j}}, \tag{1}$$

with parameter d_j we have got. Finally we accumulate all the similarities between p and all neurons given as

$$S_p=\sum_{j=1}^{m} f(d_j)=\sum_{j=1}^{m}\frac{1}{1+e^{d_j}}. \tag{2}$$

We can use the similarity S_p to decide whether the type information of p is in the information bank.

Fig. 3 shows the identifying capability of our SOM network.

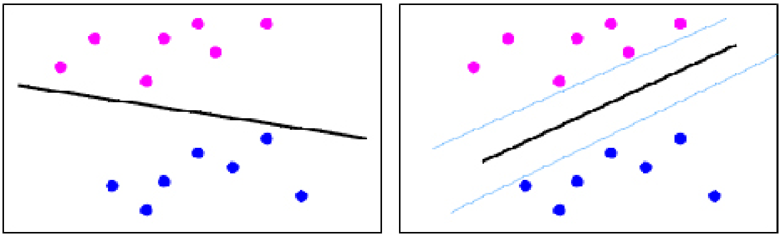


Fig. 3. This figure illustrates the SOM's capability of classification. If there are two classes to be classified, most machine learning algorithms find a separation to well classify the training data, as shown in the left image. SOM in the right image, provide the maximal capability that not only separates the training data, but has the potential to better classify the data which are not shown in the training data set.

3.2 The Module BPN

The module BPN, the second part of the machine learning engine, is designed to generate a TF for each data set, of which the structure is shown in Fig. 4. On receiving a data set from SOM, it checks additional information from the module SOM to decide whether to train the network to get a new TF for the data set. If the

SOM tells the BPN that the data set is an old one in the information bank, then what we need to do is only to get weights of the data set for the BPN from the information bank and to establish a BPN network which generates a TF to assign color and opacity to the current data set. If the information from the SOM indicates that it is a new data set, we must to train the BPN to get the weights and generate a TF for the data set, and finally the weights must be saved in the corresponding position in the information bank.

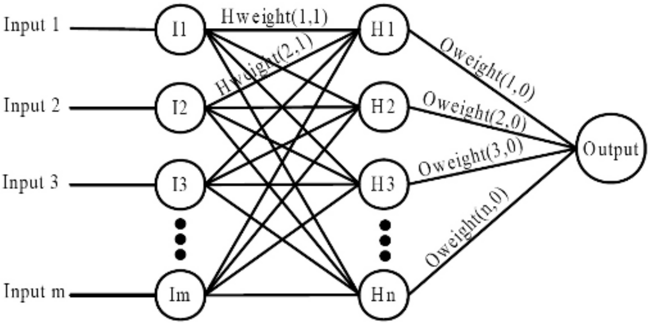


Fig. 4. Structure of an artificial neural network with *m* inputs, *n* hidden nodes, and one output

Considering the BPN network given in Fig.4, for a given sample set (X_p, Y_p) $(p = 1,2...P)$, E_p is the error function of the *p*th neuron, and the iterative weight function is $w(k+1)=w(k)+ \Delta w(k)$, where

$$\Delta w(k+1) = -\eta \frac{\partial E}{\partial w(k)}, \tag{3}$$

η is the step length,

$$\frac{\partial E}{\partial w(k)} = \sum_p \frac{\partial E_p}{\partial w(k)}, \ k = 1,2,\cdots, \tag{4}$$

k is the number of iteration.

We use two approaches to improve on BP algorithm.

(a) Adding the momentum item and decreasing item. We add the momentum item and decreasing item to $\Delta w(k+1)$, so

$$\Delta w(k+1) = -\eta \frac{\partial E}{\partial w(k)} + \alpha \cdot \Delta w(k) - \beta \cdot w(k), \tag{5}$$

where α is momentum item which stands for damp for diminishing the tendency of oscillation and better convergence, $0<=\alpha<=1$. Using a decay factor $0.01>\beta>0$ enables only those weights doing useful work in reducing the error to survive and hence improves the generalization capabilities of the network

(b) The self-adapting step length of learning (η). Because the step length of learning η is changeless on the whole learning process, the learning process will be too long or the network will not converge. So the adjusted values of connection strength and the step length of learning η should be increased when the error is increasing, and the adjusted values and the step length of learning η should be decreased when the error is decreasing. In this paper, we employ a method of self-adapting step length of learning which is given as

$$\eta = \frac{\sigma E}{\sqrt{\sum_{i \to j} (\frac{\partial E}{\partial w_{ij}})^2 + \sum_{j \to k} (\frac{\partial E}{\partial w_{jk}})^2}}, \quad (6)$$

Where σ is step length factor, $0 < \sigma < 1$.

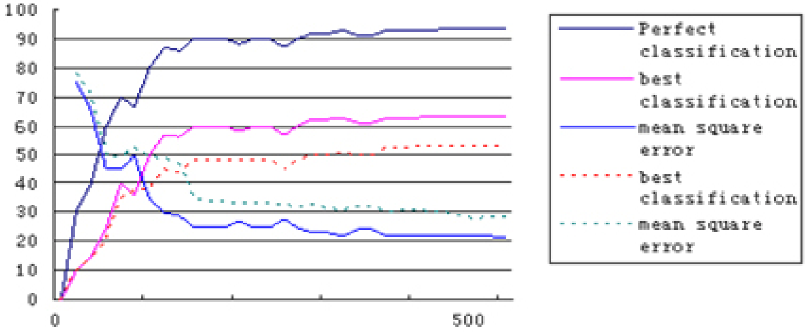
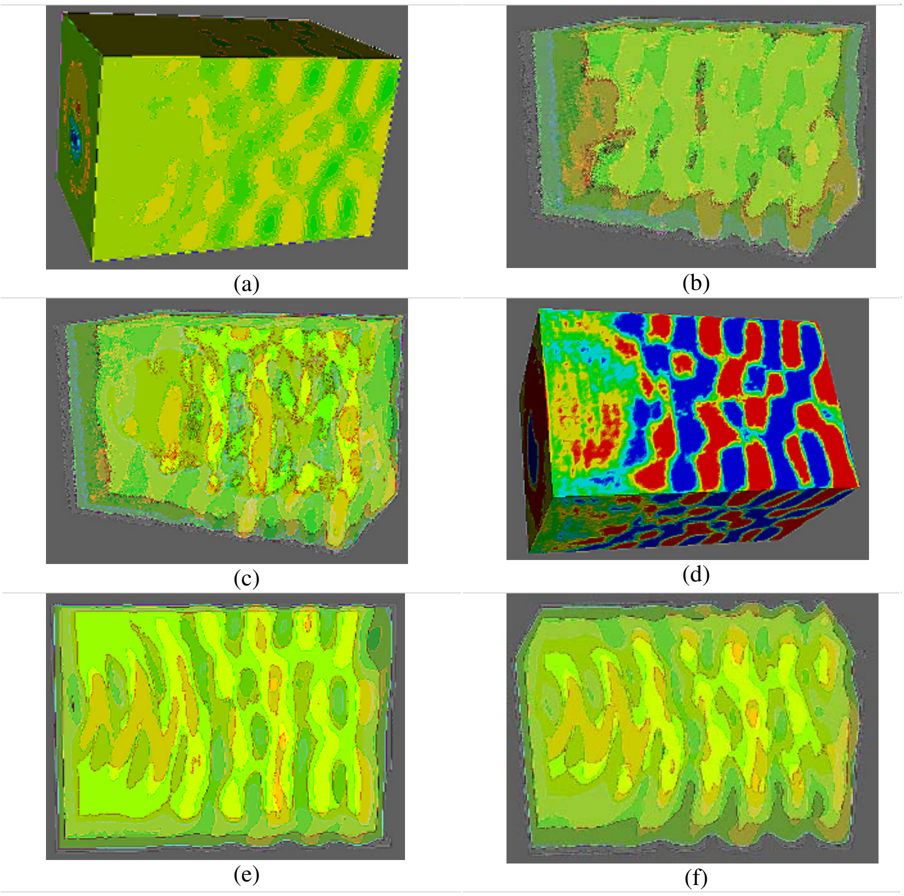


Fig. 5. Variation of perfect classification (%), best classification (%), and mean square error with number of sweeps through the training set, using a three-layered neural net with *perc (percentage of samples)* = 50, β= 0.001, α= 0.9 and m=10 nodes in each hidden layer

In Fig. 5 we illustrate the variation of the best classification and perfect classification performance and the mean square error with the number of sweeps over the training set, in which our BPN network is denoted by solid curves and the traditional BPN network is plotted using a dotted curve.

4 Experiments and Results

To test the applicability and usefulness of our proposed method, several experiments were conducted. The PC used in the experiments was with Xeon3.20GHz CPU/2.50GB memory. In the first experiment, a volume data of 100x100x300 with multiple variables was tested. This data shows the result of the simulation of a laser injecting into a plasma and interacting with each other, in which the variables are laser intensity, density and current of the plasma. Fig. 6 shows the result of the experiment. Using interactive TF without manual control or using an inappropriate TF, users can not get a good rendering result. However, with GATF, we can get a more detailed result than others.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 6. (a) The result of using surface rendering. (b) Volume rendering result of using interactive TF without manual control. (c) Volume rendering result of using interactive TF with manual control. (d) Volume rendering result with an inappropriate TF. (e) Volume rendering result of using a Trial and Error TF. (f) Volume rendering result using our GATF.

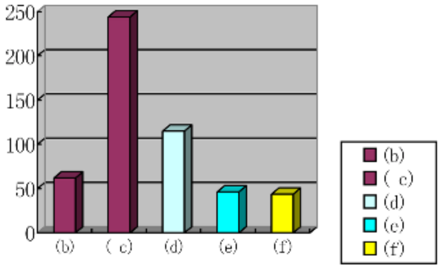
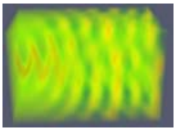
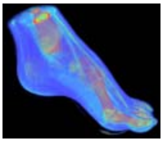
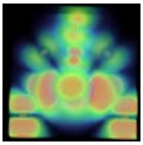


Fig. 7. Time (s) consumed in Fig. 6

Fig. 7 shows the corresponding time consuming, and we can see that GATF spends the least time rendering than others do. We also conducted other experiments with several data sets and table 1 shows the result.

Table 1. Using the Training Set Obtained Based on our approach

Data Sets	Plasma	Foot	Iron Protein
Number of Trainings Sample	391104	5038800	262144
Time of SOM Classification	2.02s	1.22s	1.05s
Time of BPN Adjusting Weights	12.50s	12.27s	12.12s
Rendered Classified Volume			

5 Conclusion and Future Work

By introducing SOM and BPN into the optimization of transfer function design, together with a proper image References evaluation strategy, a new volume rendering frameworks is proposed in this paper. Experiments results demonstrated that our method can obtain optimized transfer function and excellent rendering results. Even though it is possible to implement the training step in hardware, the current software implementation provides an adequate performance to meet the needed interactivity. In fact, a more interesting and helpful capability is fast data decompressing in hardware since one potential bottleneck for large data sets is the need to transmit data between the disk and video memory. We will explore this option in the future.

References

1. Fan Y.T., Eric B.L., Kwan L.M.: An Intelligent System Approach to Higher-Dimensional Classification of Volume Data. IEEE Trans on Visualization and Computer Graphics, 11(3) (2005) 273-284
2. Hanspeter P., Bill L., Chandrajit B., et al.: The Transfer Function Bake-Off. IEEE Computer Graphics and Applications, May/June (2001) 16-22
3. Guy M. Nicoletti: Volume visualization advances in transfer and opacity function generation for interactive direct volume rendering. IEEE Trans. on Visualization and Computer Graphics, 6(3) (2004) 124-130
4. Hladiivka, Konig A.H., and Groller E.M.: Curvature-Based Transfer Functions for Direct Volume Rendering. Spring Conference on Computer Graphics 2000 (2000) 58-65,
5. He T., Hong L., Kaufman A and Pfister H.: Generation of transfer functions with stochastic search techniques. Proceedings Visualization'96 (1996) 227-234,

6. Gordon Kindlmann and James W. Durkin: Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering. IEEE Trans. on Visualization and Computer Graphics, 11(3) (1998) 79-86
7. Shiaofen-Fang, Tom Biddlecome, and Mihran Tuceryan: Image-Based Transfer Function Design for Data Exploration in Volume Visualization. Proceedings IEEE Visualization'98 (1998) 319-326
8. http://www.terarecon.com (2006)
9. H. Pfister: Image-Centering, Using Organized Sampling. Mitsubishi Electric Research Laboratories (www.mer.com/Droiects/dg/) (2000)
10. lssei Fujishiro, Taeko Azuma, and Yuriko Takeshim: Automating Transfer Function Design for Comprehensible Volume Rendering Based on 3D Field Topology Analysis. Proceedings IEEE Visualization'99 (1999) 467-470
11. Gordon Kindlmann: Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering. Master's thesis, Cornell University (1999)
12. Gordon Kindimann and James W. Durkin: Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering. Proc. IEEE Symposium On Volume Visualization (1998) 79-86
13. Yoshinobu Sato, Carl-Fredrik Westin, Abhir Bhalerao, Shin Nakajima, Nobuyki Shiraga, Shinichi Tamura, and Ron Kikinis: Tissue Classification Based on 3D Local Intensity Structures for Volume Rendering. IEEE Transactions on Visualization and Computer Graphics (2000) 160-180
14. Vladimir Pekar, Rafael Wiemker, and Daniel Hempel: Fast Detection of Meaningful Isosurfaces for Volume Data Visualization. Proceedings IEEE Visualization 2001 (2001) 223-230
15. Jiawan Zhang, Zhigang Sun, Jizhou Sun, Zunce Wei: Moment Based Transfer Function Design for Volume rendering. International Conference on Computational Science and its Applications (ICCSA) 2003. Montreal, Canada (2003)
16. Lu H.J., R. SETIONO: Effective Data Mining Using Neural Networks. IEEE Transactions on Knowledge and Data Engineering, 8 (6) (1996) 957 – 961
17. Fan Y.T., Kwan L.M.: Intelligent Feature Extraction and Tracking for Visualizng Large-Scale 4D Flow Simulations. Super Computing'05. Seattle, Washington, USA (2005)
18. Bajaj, V. Pascucci, and D. Schicore: The Contour Spectrum. Pro. 1997 IEEE Visualization Conf.. IEEE CS Press, Los Alamitos, Ca., Oct (1997) 167-173.
19. Konig A.H., and Groller E.M.: Mastering Transfer Function Specification by Using VolumePro Technology. Proceedings of the 17th Spring Conference on Computer Graphics. (SCCG) (2001) 279-286
20. W. Schroeder, L. Sobierajski, and K. Martin, http://www.vublic.kitware.com (2006)
21. H. Hauser, et al.: Two-Level Volume Rendering. IEEE Trans. On Visualization and Computer Graphics, vo1.7, no.3, July-Sep (2001) 242-251
22. T.Kohonen: Self-Organizing Maps. 3rd edn. Springer-Verlag, Berlin (2001)

Real-Time Synthesis of 3D Animations by Learning Parametric Gaussians Using Self-Organizing Mixture Networks

Yi Wang[1], Hujun Yin[2], Li-Zhu Zhou[1], and Zhi-Qiang Liu[3]

[1] Department of Computer Science and Technology, Tsinghua University, Graduate School at Shenzhen, China
yi.wang.2005@gmail.com, dcszlz@tsinghua.edu.cn
[2] School of Electrical and Electronic Engineering, The University of Manchester, UK
h.yin@manchester.ac.uk
[3] School of Creative Media, City University of Hong Kong, Kowloon, Hong Kong
zq.liu@cityu.edu.hk

Abstract. In this paper, we present a novel real-time approach to synthesizing 3D character animations of required style by adjusting a few parameters or scratching mouse cursor. Our approach regards learning captured 3D human motions as parametric Gaussians by the self-organizing mixture network (SOMN). The learned model describes motions under the control of a vector variable called the *style variable*, and acts as a probabilistic mapping from the low-dimensional style values to high-dimensional 3D poses. We have designed a pose synthesis algorithm and developed a user friendly graphical interface to allow the users, especially animators, to easily generate poses by giving style values. We have also designed a *style-interpolation* method, which accepts a sparse sequence of key style values and interpolates it and generates a dense sequence of style values for synthesizing a segment of animation. This *key-styling* method is able to produce animations that are more realistic and natural-looking than those synthesized by the traditional key-framing technique.

1 Introduction

The traditional technique of creating 3D animations, the key-framing technique, relies heavily on intensive and expensive manual labor. In recent years, with the development of motion capture technique, which could record the 3D movement of a set of markers placed on the body of human performer, learning approaches are developed to capture characteristics of certain types of human motion and automate the synthesis of new motions according to users' requirements. Some typical and impressive works have been published on top conference and journals, include [1], [2] and [3] (c.f. Table 1).

In [1], Li and et al. used an unsupervised learning approach to learn possible recombinations of motion segments as a *segment hidden Markov model* ([4] and [5]). In [2], Grochow and et al. used a non-linear principle component analysis method called *Gaussian process latent variable model* ([6]), to project 3D poses,

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 671–678, 2006.

into a low-dimensional space called *style space*. In contrast with that as [1], the learning approach used in [2] is unsupervised, and the subject to be modelled is static poses other than dynamic motions. In [3], Brand and Hertzmann proposed to learn the human motion under control of a style variable as an improved *parametric hidden Markov model* ([7]) with an unsupervised learning algorithm ([8]). In this paper, we present a new supervised approach that learns 3D poses under control of a vector variable called *style variable*. The comparison of our approach with previous ones are listed in Table 1.

Table 1. The placement of our contribution

	Learning (dynamic) motions	Learning (static) poses
Supervised approach	[8] Brand (1999)	This paper
Unsupervised approach	[1] Li (2002); [3] Brand (2000)	[2] Grochow (2004)

The idea of extracting motion style and modeling it separately from the motion data [7] is potential to develop novel productive motion synthesis approaches that manipulate the style value other than the high-dimensional motion data. The motion data is composed of a dense sequence of 3D poses, where each pose is defined by the 3D rotations of all major joints of the human body and have to be represented by a high dimensional vector (usually over 60-D [1]). The high dimensionality makes the motion data difficult to model and to manipulate. In contrast, the style variable is usually a low-dimensional vector (2-D in our experiments) that encodes a few important aspects of the motion. These facts intrigued us to learn a probabilistic mapping from style to human motion as a conditional probabilistic distribution (p.d.f.) $P(\boldsymbol{x} \mid \boldsymbol{\theta})$, which, given a style value $\boldsymbol{\theta}$, is able to output one or more 3D poses $\boldsymbol{x}$ that have the style as specified by $\boldsymbol{\theta}$.

A well-known model that represents a conditional distribution is the parametric Gaussian, whose mean vectors are functions $f(\boldsymbol{\theta})$. However, in order to capture the complex distribution of 3D poses caused by the complex dynamics of human motion, we model $P(\boldsymbol{x} \mid \boldsymbol{\theta})$ as a mixture of parametric Gaussians. Although most mixture models are learned by the Expectation-Maximization (EM) algorithm, we derived a learning algorithm based on the self-organizing mixture network (SOMN) [9], which, different with the deterministic ascent nature of the EM algorithm, is in fact a stochastic approximation algorithm with faster convergence speed and less probability of being trapped in local optima.

2 Learning SOMN of Parametric Gaussians

The SOMN of Parametric Gaussians Model. Mixture models are a usual tool to capture complex distributions over a set of observables $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$. Denote $\boldsymbol{\Lambda}$ as the set of parameters of the model, the likelihood of a mixture model is,

$$p(\boldsymbol{x} \mid \boldsymbol{\Lambda}) = \sum_{j=1}^{K} \alpha_j p_j(\boldsymbol{x} \mid \boldsymbol{\lambda_j}) \ , \tag{1}$$

where each $p_j(\boldsymbol{x})$ is a component of the mixture, α_j is the corresponding weight of the component, and $\boldsymbol{\lambda}_j$ denotes the parameters of the j-th component.

Given the observations $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, learning a mixture model is actually an adaptive clustering process, where some of the observations, to some extent, are used to estimate a component; while others are used to estimate other components. A traditional approach for learning a mixture model is the expectation-maximization (EM) algorithm, which, as a generalization of the K-means clustering algorithm, alternatively executes two step: E-step and M-step. In the E-step each observation $\boldsymbol{x}_i$ is assigned to a component p_j to the extent λ_{ij}; and in the M-step each p_j is estimated from those observations $\boldsymbol{x}_i$ with $\lambda_{ij} > 0$. It has been proven in [10] that this iteration process is actually a deterministic ascent maximum likelihood algorithm.

The SOMN proposed by Yin and Allinson in 2001 [9] is a neural network that is a probabilistic extension of the well-known clustering algorithm, the self-organizing map (SOM), with each node representing a component of a mixture model. The main difference between the learning algorithm of the SOMN and the EM algorithm is that the former one employs the Robbins–Monro stochastic approximation method to estimate the mixture model to achieve generally faster convergence and to avoid being trapped by local optima.

In this paper, we derive a specific SOMN learning algorithm to learn the conditional probability distribution $p(\boldsymbol{x} \mid \boldsymbol{\theta})$ between 3D pose $\boldsymbol{x}$ and the motion style $\boldsymbol{\theta}$ as a mixture model of,

$$p\left(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\Lambda}\right) = \sum_{i=1}^{K} \alpha_i p_j\left(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\lambda}_i\right) \ , \tag{2}$$

where, each component $p_j(\cdot)$ a linearly parametric Gaussian distribution,

$$p_j\left(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\lambda}_j\right) = \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{W}_j\boldsymbol{\theta} + \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right) \ , \tag{3}$$

where $\boldsymbol{W}_j$ is called the *style transformation matrix*, which, together with $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ forms the parameter set $\boldsymbol{\lambda}_j$ of the j-th component.

The Learning Algorithm. Learning a SOMN of parametric Gaussians minimizes the following Kullback–Leibler divergence[1] between the true distribution $p(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\Lambda})$ and the estimated one $\hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\Lambda})$,

$$D\left(\hat{p}; p\right) = -\int \log \frac{\hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\Lambda})}{p(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\Lambda})} p(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\Lambda}) d\boldsymbol{x} \ , \tag{4}$$

[1] The Kullback–Leibler is a generalized form of the likelihood. The EM algorithm learns a model by maximizing the likelihood.

which is always a positive number and will be zero if and only if the estimated distribution is the same as the true one. When the estimated distribution is modelled as a mixture model, taking partial derivatives of Equation 4 with respect to $\boldsymbol{\lambda}_i$ and α_i leads to

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\lambda}_i} D(\hat{p}; p) &= -\int \left[\frac{1}{\hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \hat{\boldsymbol{\Lambda}})} \frac{\partial \hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \hat{\boldsymbol{\Lambda}})}{\partial \boldsymbol{\lambda}_i} \right] p(\boldsymbol{x}) d\boldsymbol{x} \ , \\
\frac{\partial}{\partial \alpha_i} D(\hat{p}; p) &= -\int \left[\frac{1}{\hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \hat{\boldsymbol{\Lambda}})} \frac{\partial \hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \hat{\boldsymbol{\Lambda}})}{\partial \alpha_i} \right] p(\boldsymbol{x}) d\boldsymbol{x} + \xi \frac{\partial}{\partial \alpha_i} \left[\sum_{j=1}^{K} \hat{\alpha}_i - 1 \right] \\
&= -\frac{1}{\hat{\alpha}_i} \int \left[\frac{\alpha_i \hat{p}_i(\boldsymbol{x} \mid \boldsymbol{\theta}, \hat{\boldsymbol{\lambda}}_i)}{\hat{p}_i(\boldsymbol{x} \mid \boldsymbol{\theta}, \hat{\boldsymbol{\Lambda}})} - \xi \hat{\alpha}_i \right] p(\boldsymbol{x}) d\boldsymbol{x} \ ,
\end{aligned} \tag{5}$$

where ξ is a Lagrange multiplier to ensure $\sum_i \alpha_i = 1$.

Following in [9], the Robbins–Monro stochastic approximation is used to solve Equation 5 because the true distribution is not known and the equation has to depend only on the estimated version. Then the following set of iterative updating rules are obtained:

$$\begin{aligned}
\hat{\boldsymbol{\lambda}}_i(t+1) &= \hat{\boldsymbol{\lambda}}_i(t) + \delta(t) \left[\frac{1}{\hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \hat{\boldsymbol{\Lambda}})} \frac{\partial \hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \hat{\boldsymbol{\Lambda}})}{\partial \boldsymbol{\lambda}_i(t)} \right] \\
&= \hat{\boldsymbol{\lambda}}_i(t) + \delta(t) \left[\frac{\alpha_i}{\hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \hat{\boldsymbol{\Lambda}})} \frac{\partial \hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \hat{\boldsymbol{\lambda}}_i)}{\partial \boldsymbol{\lambda}_{\boldsymbol{i}}(t)} \right] \ ,
\end{aligned} \tag{6}$$

$$\begin{aligned}
\hat{\alpha}_i(t+1) &= \hat{\alpha}_i(t) + \delta(t) \left[\frac{\alpha_i(t) \hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \hat{\boldsymbol{\lambda}}_i)}{\hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \hat{\boldsymbol{\Lambda}})} - \alpha_i(t) \right] \\
&= \hat{\alpha}_i(t) - \delta(t) \left[\hat{p}(i \mid \boldsymbol{x}, \boldsymbol{\theta}) - \alpha_i(t) \right] \ ,
\end{aligned} \tag{7}$$

where $\delta(t)$ is the learning rate at time step t, and $\hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\Lambda})$ is the estimated likelihood $\hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\Lambda}) \simeq \sum_i \alpha_i \hat{p}(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\lambda}_i)$. The detailed derivation of Equation 5, 6 and 7 are the same to those in [9].

To derive the partial derivative of the component distribution in Equation 6, we denote $\boldsymbol{Z}_i = [\boldsymbol{W}_i, \boldsymbol{\mu}_i]$ and $\boldsymbol{\Omega} = [\boldsymbol{\theta}, 1]^T$, so that $\hat{p}(\boldsymbol{x} \mid, \boldsymbol{\theta}, \hat{\boldsymbol{\lambda}}_i) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{W}_i \boldsymbol{\theta} + \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{Z}_i \boldsymbol{\Omega}, \boldsymbol{\Sigma}_i)$. Then, the updating rule of $\boldsymbol{Z}_i$ can be derived from Equation 6:

$$\begin{aligned}
\boldsymbol{Z}_i^{(t+1)} =& \boldsymbol{Z}_i^{(t)} + \delta(t) \left[\frac{\alpha_i}{\hat{p}(\boldsymbol{x} \mid \hat{\boldsymbol{\Lambda}})} \frac{\partial \mathcal{N}(\boldsymbol{x}; \boldsymbol{Z}_i \boldsymbol{\Omega}, \boldsymbol{\Sigma}_i)}{\partial \boldsymbol{Z}_i} \right] \\
=& \boldsymbol{Z}_i^{(t)} + \delta(t) \left[\frac{\alpha_i}{\hat{p}(\boldsymbol{x} \mid \hat{\boldsymbol{\Lambda}})} \mathcal{N}(\boldsymbol{x}; \boldsymbol{Z}_i \boldsymbol{\Omega}, \boldsymbol{\Sigma}_i) \frac{\partial \log \mathcal{N}(\boldsymbol{x}; \boldsymbol{Z}_i \boldsymbol{\Omega}, \boldsymbol{\Sigma}_i)}{\partial \boldsymbol{Z}_i} \right] \\
=& \boldsymbol{Z}_i^{(t)} + \delta(t) \left[\hat{p}(i \mid \boldsymbol{x}) \frac{\partial \log \mathcal{N}(\boldsymbol{x}; \boldsymbol{Z}_i \boldsymbol{\Omega}, \boldsymbol{\Sigma}_i)}{\partial \boldsymbol{Z}_i} \right]
\end{aligned}$$

$$
\begin{aligned}
&= \boldsymbol{Z}_i^{(t)} - \frac{1}{2}\delta(t)\hat{p}(i \mid \boldsymbol{x}) \left[\frac{\partial}{\partial \boldsymbol{Z}_i} \big(\boldsymbol{x} - \boldsymbol{Z}_i \boldsymbol{\Omega}\big)^T \Sigma^{-1} \big(\boldsymbol{x} - \boldsymbol{Z}_i \boldsymbol{\Omega}\big) \right] \\
&= \boldsymbol{Z}_i^{(t)} - \frac{1}{2}\delta(t)\hat{p}(i \mid \boldsymbol{x}) \\
&\quad \left[\frac{\partial}{\partial \boldsymbol{Z}_i} \Big(\boldsymbol{x}^T \Sigma^{-1} \boldsymbol{x} - \boldsymbol{Z}_i \boldsymbol{\Omega}^T \Sigma^{-1} \boldsymbol{x} - \boldsymbol{x}^T \Sigma^{-1} \boldsymbol{Z}_i \boldsymbol{\Omega} + \boldsymbol{Z}_i \boldsymbol{\Omega}^T \Sigma^{-1} \boldsymbol{Z}_i \boldsymbol{\Omega} \Big) \right] \\
&= \boldsymbol{Z}_i^{(t)} - \frac{1}{2}\delta(t)\hat{p}(i \mid \boldsymbol{x}) \left[\frac{\partial}{\partial \boldsymbol{Z}_i} \Big(\boldsymbol{x}^T \Sigma^{-1} \boldsymbol{x} - 2\boldsymbol{Z}_i \boldsymbol{\Omega}^T \Sigma^{-1} \boldsymbol{x} + \boldsymbol{Z}_i \boldsymbol{\Omega}^T \Sigma^{-1} \boldsymbol{Z}_i \boldsymbol{\Omega} \Big) \right] \\
&= \boldsymbol{Z}_i^{(t)} - \frac{1}{2}\delta(t)\hat{p}(i \mid \boldsymbol{x}) \left[-2 \frac{\partial}{\partial \boldsymbol{Z}_i} (\boldsymbol{Z}_i \Omega)^T \Sigma^{-1} \boldsymbol{x} + \frac{\partial}{\partial \boldsymbol{Z}_i} (\boldsymbol{Z}_i \Omega)^T \Sigma^{-1} \boldsymbol{Z}_i \Omega \right] \\
&= \boldsymbol{Z}_i^{(t)} - \frac{1}{2}\delta(t)\hat{p}(i \mid \boldsymbol{x}) \left[-2 \frac{\partial}{\partial \boldsymbol{Z}_i} \Omega^T \boldsymbol{Z}_i^T \Sigma^{-1} \boldsymbol{x} + \frac{\partial}{\partial \boldsymbol{Z}_i} (\Omega^T \boldsymbol{Z}_i^T \Sigma^{-1})(\boldsymbol{Z}_i \Omega) \right] \\
&= \boldsymbol{Z}_i^{(t)} - \frac{1}{2}\delta(t)\hat{p}(i \mid \boldsymbol{x}) \Sigma^{-1} \left[\boldsymbol{x}\Omega^T - Z\Omega\Omega^T \right] \ . \qquad (8)
\end{aligned}
$$

So, the updating rule of $\boldsymbol{Z}_i$ is,

$$
\Delta \boldsymbol{Z}_i = -\frac{1}{2}\delta(t)\hat{p}(i \mid \boldsymbol{x}) \Sigma^{-1} \left[\boldsymbol{x}\Omega^T - Z\Omega\Omega^T \right] \ . \qquad (9)
$$

By considering $\hat{p}(i \mid \boldsymbol{x}, \boldsymbol{\theta})$, which is a Gaussian function, as the Gaussian neighborhood function, we can consider Equation 9 exactly as the SOM updating algorithm. Although an updating rule of $\Delta \boldsymbol{\Sigma}_i$ may be derived similarly, it is unnecessary in the learning algorithm, because the covariance of each component distribution implicitly corresponds to the neighborhood function $\hat{p}(i \mid \boldsymbol{x})$, or, the spread range of updating a winner at each iteration. As the neighborhood function has the same form for every nodes, the learned mixture distribution is homoscedastic.

3 SOMN of Parametric Gaussians for Motion Synthesis

Determine the Physical Meaning of the Style Variable. A learned SOMN of parametric Gaussian model $p(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\Lambda})$ could be considered as a probabilistic mapping from a given style value $\hat{\boldsymbol{\theta}}$ to a 3D poses $\hat{\boldsymbol{x}}$. If the users know the physical meaning of each dimension of the style variable $\boldsymbol{\theta}$, they can give precise style value $\hat{\boldsymbol{\theta}}$ to express their requirement to the synthesized poses. The supervised learning framework presented in Section 2 allows the users to determine physical meaning of the style variable prior to learning.

As an example, suppose that we captured a boxing motion as training data, where the boxer sometimes crouches to evade from attacking and some other times punches his fist to attack. We can use a 2-dimensional style variable to describe the details of the boxing motion, where one dimension encodes the body height, which varies from crouching to standing up, and with the other dimension encodes the distance of arm when punching. Once the physical meaning of each dimension of the style variable is determined, the style values $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N\}$ of each one of the training frames $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ can be calculated from the training motion itself.

It is notable that if we carefully choose a number of dimensions of the style variable that encode visually independent characteristics of the training motion, the *style space*, which is spanned by all possible style values, will be an Euclidean space, within which, any curve corresponds to a smooth change of the style value. This is interesting for synthesizing character animations, instead of static poses, because the smooth change of motion style like body height and punch distance usually leads to smooth body movement. Experiments are shown in Section 4.

Generate 3D Pose from Given Style Value. Given a learned SOMN of parametric Gaussians $p(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{\Lambda})$ with K components, mapping a given style value $\hat{\boldsymbol{\theta}}$ to a 3D pose $\hat{\boldsymbol{x}}$ can be achieved by substitute $\hat{\boldsymbol{\theta}}$ into the model and draw a sample $\hat{\boldsymbol{x}}$ from the distribution $p(\boldsymbol{x} \mid \hat{\boldsymbol{\theta}}, \boldsymbol{\Lambda})$. Although the Monte Carlo sampling method is generally applicable for most complex distributions, to avoid the intensive computation and achieve real-time performance, we designed the following two step algorithm as shown in Algorithm 1 to calculate the pose $\hat{\boldsymbol{x}}$ with the highest probability. The first step of the algorithm calculate the poses $\{\hat{\boldsymbol{x}}_j\}_{j=1}^K$ that are most probable for each component p_j of the learned model; and then the algorithm selects and returns the most probable one $\hat{\boldsymbol{x}}$ among all the $\{\hat{\boldsymbol{x}}_j\}_{j=1}^K$.

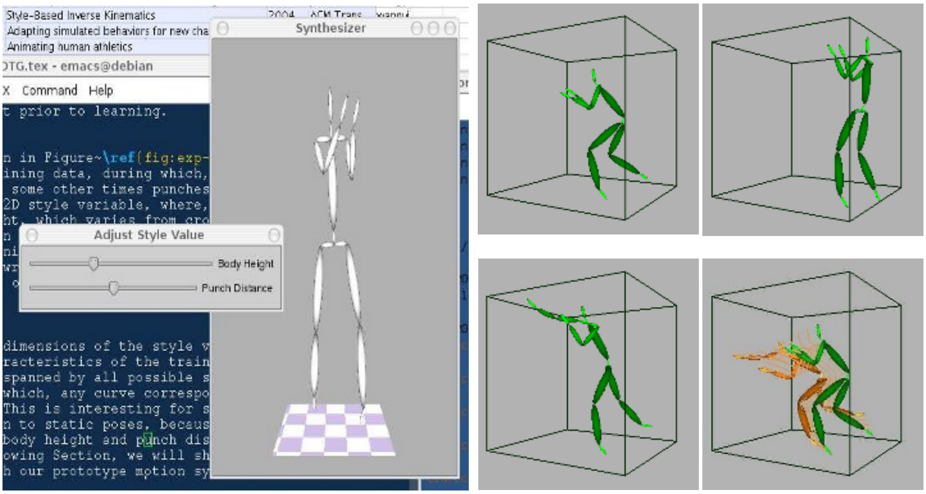
```
input  : The given new style $\hat{\boldsymbol{\theta}}$
output : The synthesized pose $\hat{\boldsymbol{x}}$
calculate the most probable pose from each component;
foreach $j \in [1, K]$ do
    $\hat{\boldsymbol{x}}_j \leftarrow \boldsymbol{W}_j \hat{\boldsymbol{\theta}} + \boldsymbol{\mu}_j$;
end
select the most probable one among the calculation result;
$j \leftarrow \operatorname{argmax}_j \alpha_j p_j(\hat{\boldsymbol{x}}_j \mid \hat{\boldsymbol{\theta}}, \boldsymbol{\Lambda})$;
$\hat{\boldsymbol{x}} \leftarrow \hat{\boldsymbol{x}}_j$;
```

Algorithm 1. synthesize pose from given style value

The Prototype of Motion Synthesis System. We developed an interactive graphical user interface (GUI) program as shown in Figure 1 to ease the pose and motion synthesis. With the parameter adjustment panel (to the left of the main window), users are able to specify a style value by adjusting every dimension of the style variable. The changed style value is instantly input to Algorithm 1, and the synthesized pose $\hat{\boldsymbol{x}}$ is displayed in real-time.

With this GUI program, users can also create animations by (1) select a sparse sequence of key-styles to define the basic movement of a motion segment, (2) produce a dense sequence of style values interpolating the key-styles, and (3) map each style value into a frame to synthesize the motion sequence. As the traditional method of producing character animations is called *keyframing*, which interpolate a sparse sequence of keyframes, we name our method *key-styling*.

A known problem of keyframing is that the synthesized animation seems rigid and robotic. This is because the keyframes is represented by a high-dimensional vector consisting of 3D joint rotations. Evenly interpolating the rotations cannot



```
t prior to learning.

n in Figure~\ref{fig:exp-
ining data, during which,
 some other times punches
2D style variable, where,
ht, which varies from cro
n
ni
wr
 o

dimensions of the style v
racteristics of the train
spanned by all possible s
which, any curve correspo
This is interesting for s
n to static poses, becaus
body height and punch dis
owing Section, we will sh
h our prototype motion sy
```

Fig. 1. The prototype motion synthesis system

ensure evenly interpolated dynamics. While, interpolating the key-styles results in smooth change of the major dynamics, and style-to-pose mapping adds kinematics details to the motion. The change of kinematics details does not need to be evenly.

4 Experiments

To demonstrate the usability of our synthesis approach, we captured a segment of boxing motion of about 3 minutes under the frame-rate of 66 frame-per-second as the training data. Some typical poses in the motion is shown in Figure 1 (a), (b) and (c). Because the boxer sometimes crouches to evade and some other times punches his fist to attack, we use a 2-dimensional style variable to encode the body height and the distance of punching.

Once the dimensionality of style variable is determined, labelling the training frames with style values is not a difficult problem. For the application of automatic motion synthesis, we must have the skeleton (the connections of joints) for rendering the synthesized motion and must have the rotations of joints as training data. With these two kinds of informations, it is easy to compute the style value $\boldsymbol{\theta}_i$ for each training frame $\boldsymbol{x}_i$. In our experiment, we wrote a simple Perl script program to calculate the 3D positions of the joints and to derive the style values.

After estimating a SOMN of parametric Gaussians from the labelled training frames, we can give new style value by dragging the slide bars of our prototype motion synthesis system (as shown in Figure 1). A simple dragging of the slide bar that represents the punch distance synthesized a segment of animation as shown in Figure 1(d).

5 Conclusion and Discussion

In this paper, we present a novel approach to real-time synthesis of 3D character animations. The first step of the approach is to learn a probabilistic mapping from a low-dimensional style variable to high-dimensional 3D poses. By modeling the probabilistic mapping using the SOMN on parametric Gaussians, we come up a learning algorithm which is numerically more tolerant to local optima problem and converges faster than the EM-based algorithms for learning mixture models. The supervised learning frame gives the users an interface to specify the physical meaning of each dimension of the style variable. So, given a learned model and using our prototype motion synthesis system, the users are able to create 3D poses by simply dragging slide-bar widgets and/or to produce desired character animations by the so-called *key-styling* method.

Acknowledgment

We thank Microsoft Research Asia for providing part of the motion capture data used in our experiments. This work is supported in part by Hong Kong RGC CERG CityU1247/03E.

References

1. Li, Y., Wang, T., Shum, H.Y.: Motion texture: A two-level statistical model for character motion synthesis. Proc. ACM SIGGRAPH (2002) 465–472
2. Grochow, K., Martin, S.L., Hertzmann, A., Popović, Z.: Style-based inverse kinematics. In: Proc. ACM SIGGRAPH. (2004) 522 – 531
3. Brand, M., Hertzmann, A.: Style machines. In: Proc. ACM SIGGRAPH. (2000) 183–192
4. Ostendorf, M., Digalakis, V.V., Kimball, O.A.: From HMM's to segment models: a unified view of stochastic modeling for speech recognition. IEEE Transactions on Speech and Audio Processing **4(5)** (1996) 360–378
5. Gales, M., Young, S.: The theory of segmental hidden Markov models. Technical report, Cambridge Univ. Eng. Dept. (1993)
6. Lawrence, N.D.: Gaussian process latent variable models for visualisation of high dimensional data. In: Proc. 16th NIPS. (2004)
7. Wilson, A.D., Bobick, A.: Parametric hidden markov models for gesture recognition. IEEE Trans. Pattern Analysis Machine Intelligence **21(9)** (1999) 884–900
8. Brand, M.: Pattern discovery via entropy minimization. In Heckerman, D., Whittaker, C., eds.: Artificial Intelligence and Statistics, Vol. 7. Volume 7. Morgan Kaufmann, Los Altos (1999)
9. Yin, H., Allinson, N.M.: Self-organizing mixture networks for probability density estimation. IEEE Trans. Neural Networks **12** (2001) 405–411
10. Ormoneit, D., Tresp, V.: Averaging, maximum penalised likelihood and bayesian estimation for improving gaussian mixture probability density estimates. IEEE Trans. Neural Networks **9** (1998) 639–650

An Excellent Feature Selection Model Using Gradient-Based and Point Injection Techniques

D. Huang and Tommy W.S. Chow

Department of Electrical Engineering, City University of Hong Kong
dihuang@cityu.edu.hk, eetchow@cityu.edu.hk

Abstract. This paper focuses on enhancing the effectiveness of filter feature selection models from two aspects. One is to modify feature searching engines based on optimization theory, and the other is to improve the regularization capability using point injection techniques. The second topic is undoubtedly important in the situations where overfitting is likely to be met, for example, the ones with only small sample sets available. Synthetic and real data are used to demonstrate the contribution of our proposed strategies.

1 Introduction

As computer technology advances rapidly, data are accumulated in an enormous speed unprecedentedly experienced in human history. In some advanced engineering and physical science applications, most conventional computational methods have already experienced difficulty in handling the enormous data size. In handling these data sets, feature selection is an essential and widely used technique. It reduces the size of features through eliminating irrelevant and redundant features, and thus results with increased accuracy, enhanced efficiency, and improved scalability for classification and other applications such as data mining (Han, 2001). Feature selection is especially important when one is handling a huge data set with dimensions up to thousands.

A feature selection framework generally consists of two parts: a searching engine used to determine the promising feature subset candidates and a criterion used to determine the best candidate (Liu, 1998; Molina, 2002). Currently, there are several searching engines: ranking, optimal searching, heuristic searching and stochastic searching. Among these engines, heuristic searching, which can easily be implemented and is able to deliver respectable results (Pudil, 1994), is widely used. Feature selection models can be broadly categorized as filter model, wrapper model, and embedded model according to their evaluation criteria. Filter models explore various types of statistical information, such as distribution probabilities underlying data. Wrapper and embedded models are classifier-specified and the selected features may vary with different classifiers. Given a feature subset, say *S*, wrapper and embedded models firstly require to build a classifier based on *S*. Wrapper models then rely on the performance of the built classifier to determine the goodness of *S*, while embedded models make use of the parameters of the built classifier to assess *S*. Wrapper models are usually more computationally expensive than filter models.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 679 – 692, 2006.

In a filter model, good feature selection results rely on a respectable evaluation criterion and an appropriate searching strategy. The former issue has been heavily investigated. Various types of information, including mutual information (Battit, 1994; Bonnlander, 1996; Chow, 2005), correlation (Hall 1999), etc., have been explored for evaluating features. By comparison, there are fewer studies focused on searching engines. Also, most of those studies are completely designed in discrete feature domains. For example, sequential forward searching (SFS), a typical heuristic searching scheme, identifies k important features from unselected features and places them into a selected feature subset in each iteration. To improve SFS, a stepwise strategy is designed – in each iteration, selecting k "good" unselected features is followed by deleting r "worst" selected features ($r < k$) (Pudil, 1994). And Al-Ani *et al.*, (2000) employ "elite" selected features, not all of them, to identify important features from unselected ones. These algorithms, studied in a discrete feature space, depend on the testing of more feature combinations in order to deliver improved results. Clearly, more testing will increase the computational complexity.

Given a set of n samples $D = \{(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)\}$ which is drawn from a joint distribution P on $X\times Y$, a feature selection process is per se a learning process in the domain of $X\times Y$ to optimize the employed feature evaluation criterion, say $L_{(x,y)\sim P}(x, y)$. As P is unknown, $L_{(x,y)\sim P}(x, y)$ has to be substituted by $L_{(x,y)\in D}(x, y)$. Clearly, when D cannot correctly represent P, this substitution may cause overfitting in which the selected features are unable to deal with testing data satisfactorily despite performing splendidly on the training data D (Bishop, 1995). In many applications, a machine learning process suffers from insufficient learning samples. For instance, in most microarray gene profile expression based cancer diagnosis data sets that may only consist of tens samples. With small sample sets, overfitting is likely to happen and this issue must be addressed accordingly. A wrapper/embedded feature selection models always involve with classification learning processes. Thus, the regularization techniques developed for classification learning can be directly employed in a wrapper/embedded model. For example, support vector machine and penalized Cox regression model, which have been argued to have high generalization capability, are employed in embedded models (Guyon, 2002; Gui, 2005). And an embedded feature selection model is trained based upon with regularized classification loss functions (Perkins, 2003). On the other hand, since filter schemes do not explicitly include a classification learning process, the regularization techniques developed for classification learning cannot be explored. In this sense, it is a need to design specified regularization strategies. To our knowledge, only a few attempts are done on this topic. In order to address problem of overfitting, a bootstrap framework has been adopted for mutual information estimation (Zhou, 2003). Under this framework, mutual information estimation should be conducted several times in order to deliver one result. The bootstrap framework is thus highly computationally demanding, which precludes it from being widely used.

In this paper, we propose two strategies – the one is for improving the effectiveness of searching engines, and the other is for addressing the problem of overfitting. And we choose a typical filter feature selection model as example to demonstrate these strategies. In this filter model, the searching engine and the feature evaluation index are SFS (Devijver, 1982) and Bayesian discriminant criterion (BD) (Huang, 2005), respectively. We firstly analyze SFS according to the well-established optimization theory (Bishop, 1995). The analysis of this type, which has been overlooked in previous studies, can reveal the shortcoming of conventional SFS – SFS is unable to perform optimization in a maximal way. To address this issue, we naturally come to the optimization theory. As a result, a modified SFS is proposed, which conducts the feature searching along the possible steepest optimization direction. To enhance the regularization capability, a point injection approach is proposed. This approach generates certain points according to the distribution of given samples, which is similar to the ones developed for classification learning. In our proposed approach, the injected points are just employed for evaluating the feature subsets. This mechanism is able to minimize the undesired side-effect of injected points.

In the next section, the *BD* sequential forward searching (SFS) feature selection model is briefed. After that, our proposed strategies are described in section 3. Finally the proposed strategies are extensively evaluated.

2 Bayesian Discriminat Based Sequential Forward Feature Searching Process

Assume that the feature set of n-sample dataset D is $F = \{f_1, f_2, \ldots, f_M\}$. Also, each pattern (say, x_i) falls into one of L categories, i.e., $y_i = \omega_k$ where $1 \le i \le n$ and $1 \le k \le L$.

2.1 Bayesian Discriminant Feature Evaluation

In filter models, probability based feature evaluation criteria are commonly used. Bayesian discriminant criterion (BD), a typical probability based approach, is developed by Huang *et al.* (2005). With the dataset D, BD is defined as

$$BD(S) = \frac{1}{n}\sum_{i=1}^{n} \log\frac{p_S(y_i \mid x_i)}{p_S(\bar{y}_i \mid x_i)} = \frac{1}{n}\sum_{i=1}^{n} \log\frac{p_S(y_i \mid x_i)}{1 - p_S(y_i \mid x_i)}, \tag{1}$$

where $\bar{y}_i$ means all the classes but class y_i, and $p_S(.)$ represents a probability which is estimated in the data domain defined by S. As shown in (1), $BD(S)$ directly measures the likelihood of given samples being correctly recognized in the data domain defined by S. A large BD(S), which indicates that most given samples can be correctly classified, is preferred.

And in our study, the probabilities required by $BD(S)$ are estimated with Parzen window (Parzen, 1962) which is modeled as

$$p(x, y) = \sum\nolimits_{all\,(x_i, y_i) \in \mathrm{class}\; y} p(x_i)\, p(x \mid x_i) = \sum\nolimits_{y_i = y} p(x_i)\, \kappa(x - x_i, h_i)\,, \tag{2}$$

$$p(x) = \sum\nolimits_{all\; c\mathrm{lasses}} p(x, y) = \sum\nolimits_{all(x_i, y_i) \in D} p(x_i)\, \kappa(x - x_i, h_i)\,, \tag{3}$$

where κ and h_i are the kernel function and the width of window, respectively. The parzen window estimator (2) or (3) has been shown to be able to converge the real probability when κ and h_i are selected properly (Parzen, 1962). κ is required to be a finite-value nonnegative function and satisfies $\int \kappa(x - x_i, h_i) dx = 1$. And the width of

κ , i.e. h_i, is required to have $\lim_{n\to\infty} h = 0$ where n is the number of given samples. Following the common way, we choose Gaussian function as κ. That is,

$$\kappa(x - x_{\mathrm{i}}, h_i) = G(x - x_i, h_i) = \frac{1}{(2\pi h_i^2)^{M/2}} \exp\left(-\frac{1}{2h_i^2}(x - x_i)(x - x_i)^T \right),$$

where M is the dimension of x. And the window width h_i is set with h_i=2*distance*(x_i,x_j) where x_j is the 3rd nearest neighbor of x_i. We use Euclidean distance, i.e., $\mathrm{distance}(x_i, x_j) = \sqrt{(x_i - x_j)(x_i - x_j)^T}$ for two data vectors x_i and x_j. As to $p(x_i)$ of the equations (2) and (3), it is estimated with $p(x_i) = 1/n$. With the equations (2) and (3) and based on p(y|x)=p(x,y)/p(x), p(y|x) required by *BD(S)* is finally obtained.

2.2 Sequential Forward Searching

In a BD based feature selection process, the aim is to determine the feature subset *S* that can maximize *BD(S)* (1). In general, *BD*(*S*) is optimized in the following way: after a pool of feature subsets is suggested by a searching engine, BD of each suggested feature subset is calculated, and one with the largest BD is either outputted as the finial feature selection result or remembered as the reference to guide the subsequent feature selection process. Many schemes for determining feature subset pools have been developed to trade the quality of optimization results with computational consumption. Among these schemes, the sequential forward searching (SFS) is the most popular one.

The SFS firstly sets the selected feature set (denoted by *S*, below) empty and enriches *S* through iteratively adding *k* important features into it. In each iteration, to select the *k* features, all the feature combinations {*S*, *k* unselected features} are examined, and the one with the largest BD is selected out to remember as a new *S*. Based upon this *S*, another iteration of feature selection is conducted. This process continues until certain stopping criteria are met.

3 Modified Sequential Forward Searching Scheme

3.1 Weighting-Sample

The objective of feature selection is to optimize the employed evaluation criterion, for example, *BD*(*S*) (1) in this study, through adjusting *S*. To clearly explain our idea, we recast *BD*(*S*) (1) as

$$BD(S) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\log \frac{p_S(y_i \mid x_i)}{1 - p_S(y_i \mid x_i)}}_{f((x_i, y_j), S)} = \frac{1}{n}\sum_{i=1}^{n} \log f((x_i, y_i), S) . \qquad (4)$$

According to the optimization theory, the steepest direction of adjusting *S* to maximize (4) is determined by

$$\frac{\partial BD(S)}{\partial S}=\frac{1}{n}\sum_{i=1}^{n}\frac{\partial BD(S)}{\partial f((x,y),S)}\frac{\partial f((x,y),S)}{\partial S}\Bigg|_{(x_i,y_i)} \tag{5}$$

It shows that, to optimize $BD(S)$, the updating of S depends on the two terms, $\partial BD(S)/\partial f((x,y),S)$ and $\partial f((x,y),S)/\partial S$. The former one happens in a continuous domain, while the latter one is related to S and has to be tackled in a discrete feature domain. In this sense, (5) cannot be solved directly. To maximize $BD(S)$, SFS tests all combinations of S and an unselected feature, and remains the one having the maximal BD. Clearly, SFS only considers the second term of (5), but overlooks the first term. It means that the searching direction of SFS is not in accordance with the steepest optimization one. This shortcoming may reduce the optimization effectiveness, and thus motivates our modification.

Naturally, our proposed strategy is based on the optimization theory, i.e., equation (5). The second term of (5) is resolved by using any conventional discrete-domain searching scheme. We use SFS for this purpose. The first term of (5) can be directly calculated in the way of

$$\frac{\partial BD(S)}{\partial f((x,y),S)}=\frac{\partial \log(f(x,y),S)}{\partial f((x,y),S)}=\frac{1}{f((x,y),S)}=\frac{1-p_S(y\,|\,x)}{p_S(y\,|\,x)}. \tag{6}$$

This shows that $\partial BD(S)/\partial f((x,y),S)$, which is only related to x, is independent of the change making on S. With this observation, we use (6) as weights to samples. In such way, feature searching is conducted with the weighted samples, not the original ones.

Assume that the dataset D is weighted by $\{w_1,w_2,\ldots,w_n\}$. With this weighted dataset, the criterion BD (1) and the probability estimations (2) and (3) are adjusted accordingly. The rule of $p(x_i)=1/n$ is replaced by $p(x_i)=w_i/n$. Also, we have

$$p(x,y)=\sum\nolimits_{all\,(x_i,y_i)\in\text{class }y}\frac{w_i}{n}G(x-x_i,h_i). \tag{7}$$

And the criterion BD is modified as

$$BD(S)=\frac{1}{n}\sum\nolimits_{i=1}^{n}w_i\log\frac{p_S(y_i\,|\,x_i)}{1-p_S(y_i\,|\,x_i)}=\frac{1}{n}\sum\nolimits_{i=1}^{n}w_i\log\frac{p_S(x_i,y_i)}{\sum\nolimits_{all\;y_j\neq y_i}p_S(x_i,y_j)}. \tag{8}$$

Apparently, it is natural to regard different samples may have different contributions to the learning processes. Currently, most machine learning algorithms have already incorporated this idea. For instance, the classification learning aims to minimize the mean square error $L(\Lambda)=\sum\nolimits_{\text{all }(x_i,y_i)}(f(x_i,\Lambda)-y_i)^2$ by training the model f, i.e., adjusting the parameter set $\wedge$ of f. The steepest decent type algorithm, which is commonly used for classification learning, determines the updating direction with

$$-\frac{\partial E}{\partial \Lambda}=\sum\nolimits_{\text{all }(x_i,y_i)}-\frac{\partial E}{\partial f}\frac{\partial f(x,\Lambda)}{\partial \Lambda}\Bigg|_{x=x_i,y=y_i}=\sum\nolimits_{\text{all }(x_i,y_i)}-\big(f(x,\Lambda)-y\big)\frac{\partial f(x,\Lambda)}{\partial \Lambda}\Bigg|_{x=x_i,y=y_i}, \tag{9}$$

where (x_i,y_i) is a given training sample. It is noted that the contribution of (x_i,y_i) is penalized by $|f(x_i,\wedge)-y_i|$. Another example is AdaBoosting (Hastie et al., 2001), a typical boosting learning algorithm. During the course of learning, AdaBoosting repeats weighting the sample (x_i,y_i) with $w_i e^{-y_i f(x_i)}$ where w_i is the current weight to (x_i,y_i). Also, in order to reduce the risk of overfitting, it is intuitively expected that the negative samples (i.e., incorrectly-recognized ones) have more influence to the subsequent learning than positive ones do. In such a way, the convergence rate can be speeded up, and the problem of overfitting can be alleviated (Lampariello *et al.*, 2001). AdaBoosting clearly can meet this expectation. The equation (9), however, indicates that the steepest decent algorithm fell short on tackling overfitting in a way that the correctly-recognized patterns still carry large weights. This fact has motivated modifications on the gradient-based algorithms (Lampariello *et al.*, 2001). Consider our proposed weighting-sample strategy, defined by equation (6). It penalizes the negative patterns heavily. Thus it will be helpful in alleviating the problem of overfitting.

3.2 Point Injection

Overfitting is caused by the deviation between the real optimization goal and the actual achievable optimization objective. The real goal of the BD based feature selection process is to maximize $BD_P(S)$ where P is the underlying probability. Since P is unknown in most cases, $BD_P(S)$ can not be actually defined, and thus has to be substituted with its empirical estimate $BD_D(S)$ (simplified as $BD(S)$, like equation (1) does). When $BD(S)$ cannot always reflect $BD_P(S)$ correctly, overfitting is caused. To avoid overfitting, it is preferred that $BD_P(S)$ varies smoothly enough.

In the area of classification/regression, overfitting can be tackled through modifying the employed empirical objective function with regularization terms. These regularization terms penalized the complex models. With them, the simple learned models can be obtained, and the likelihood of overfitting happens will thus be decreased (Bishop, 1995). The penalty terms, however, cannot always be built without thorough theoretical analysis. This is especially the case when the parameters or factors controlling smoothness of a training model are hard to determine. Another widely used regularization technique is point injection. It is known that smooth means that samples near to each other should correspond to similar performance, which is the rationale behind the techniques of point injection. In many literatures, this technique is referred as *noise injection* (Matsuoka 1992; Skurichina *et al.*, 2000; Zagoruiko *et al.*, 1976), but it is certainly expected that injected points are not real noise. Thus, to avoid the confusion, we use the term *point injection* instead of *noise injection* in this paper.

Under the frameworks of classification/regression learning, injected points are always treated just like the original samples – a classifier/regression model is built upon the original samples as well as the injected points. This working mechanism requires high-quality points. Spherical Gaussian distributed points are generated around each training object (Bishop, 1995; Matsuoka 1992). Then, the undesirable fact that the added points may increase the complexity of the solved problem is revealed. To avoid this, high quality injected points, such as, k-NN direction points (Skurichina *et al.*, 2000) and eigenvector direction points (Zagoruiko *et al.*, 1976), are suggested to replace Gaussian distributed points. Also, points are generated in a way of feature-knock-out (Wolf *et al.*, 2004). With the injected points of improved quality, contributions of injected point techniques are naturally enhanced. In this study, we reduce the risk caused by point injection through adopting a different working mechanism. Under our mechanism, only the given samples are used for building the probability estimators required by our feature evaluation criterion BD, and the given samples as well as the injected points are employed for evaluating feature subsets. Without participating in the process of model-building, the undesirable impacts of injected points must be reduced.

Around a pattern x_i, a point injection technique adds v points which are generated from a distribution $b(x\text{-}x_i)$. v and $b(x\text{-}x_i)$ play important parts in a point injection scheme (Kim, 2002; Skurichina, 2000). In order to strike the balance between performance stability and computational efficiency, v can be determined. Also, it has been argued that, for the reasonable choice of v, such as $v = 8$, 10 or 20, the effect of point injection is slightly different (Skurichina, 2000). We thus set $v = 10$. As to $b(x\text{-}x_i)$, the "width" of $b(x\text{-}x_i)$, which determines the variance of the injected points, is crucial. Since the aim of point injection is to test the properties of the region around x_i, a large width of $b(x\text{-}x_i)$ is not expected. And a small width of $b(x\text{-}x_i)$ must correspond to the insignificant contribution.

To determine an appropriate width, the simulation based strategies can be used (Skurichina, 2000). We develop an analytic approach to determine the width of $b(x\text{-}x_i)$. This approach is inspired by the ideas mentioned in (Glick, 1985; Kim, 2002). Aiming to reduce the bias intrinsic to the re-substitution error estimation as much as possible (Glick, 1985), our approach depends on the joint distribution (X,C) to determine the width of $b(x\text{-}x_i)$. Around a given pattern, say x_i, we generate several points around from Gaussian distribution $N(x_i,\sigma_i)$ where $\sigma_i = d_i/2$ and d_i is the distance of x_i to the nearest samples, i.e.,

$$d_i = \arg \min_{j,\, j\neq i} \left\| x_i - x_j \right\| . \tag{10}$$

In this way, it can be guaranteed that x' having $\|x_i\text{-}x'\| = d_i$ occurs with the close-zero probability.

The given sample set D cannot cover each part of the whole data domain very well. In turn, the probability estimators built with these samples cannot describe every part of the data domain. In detail, there may exist the parts where the conditional probabilities $p(x|y)$ for all classes are very small. According to the equation (8), it is that $\left| \frac{\partial BD(S)}{\partial p(x,\omega)} \right| \propto \frac{1}{p(x,\omega)}$ for all classes ω. It indicates that, when all $p(x|y)$ are small, a very little change of x will cause a large change of $BD(S)$. The points of such type are not expected.

For the originally given samples on which the probability models are built, at least one $p(x|y)$ must be large enough. On the other hand, an injected point may be uncertain. That is, all the probabilities about it are very small. It is better to minimize the impact of uncertain points, although it can be argued that they may equally affect the quality of different feature subset candidates. With this idea, the way of calculating *BD*(*S*) of injected points is modified. Suppose that, according to the given *D*, we generate dataset *D'* for which we have

$$BD(S)\big|_{D'} = \frac{1}{|D'|} \sum_{\text{all } (x'_i, y_i') \in D'} w'_i \log \frac{p_S(y'_i \mid x'_i)}{p_S(\overline{y'_i} \mid x'_i)} \underbrace{\left(\arg\max_{all\ \omega} (p_S(x'_i \mid \omega)) \right)}_{A} \tag{11}$$

where |*D'*| means the cardinality of *D'*. y'_i and w'_i are the weight and class label of x'_i and are inherited from the corresponding sample in *D*. With part A, the impact of uncertain points will be limited, which satisfies our expectation.

Below, contributions of the point injection strategy are assessed on a group of 3-class and 8-feature synthetic datasets. In these data, the first four features are generated according to

Class 1 ~ *m* samples from *N*((1, 1, -1, -1), σ),
Class 2 ~ *m* samples from *N*((-1, -1, 1, 1), σ),
Class 3 ~ *m* samples from *N*((1, -1, 1, -1), σ).

And the other four features are randomly determined from normal distribution with zero means and unit variance. Clearly, among totally eight features, the first four are equally relevant to the classification task, and the others are irrelevant. Three feature selection methods are applied to this data to determine four salience features. They are the conventional SFS, SFS with the feature-knock-out and the proposed point injection approaches. Only if all relevant features are selected out, the selection results can be considered correct.

Table 1. Comparisons on a synthetic data. These results demonstrate the merits of the proposed point injection strategy.

		SFS	SFS with feature-knock-out strategy	SFS with the point injection strategy
σ = 0.3	m = 3	0.980	0.941	0.991
	m = 9	1.000	1.000	1.000
σ = 0.8	m = 3	0.357	0.358	0.392
	m = 9	0.933	0.930	0.931

Different settings of σ and *m* are investigated. For reliable estimation, in each setting, three feature selection methods are run on 10,000 datasets independently generated. And the correct results over 10,000 trials are counted. In Table 1, the correctness ratios are presented. It shows that the feature-knock-out point injection strategy cannot bring the improved feature selection results in this example. This may be because this strategy is originally designed for classification learning, not for feature selection. Turn to the proposed point injection strategy. Its advantage becomes more significant either when the sample size becomes small or when σ becomes large. All these conditions actually mean there is a high likelihood of overfitting since a larger σ means a more complex problem. Thus, the presented results suggest that our approach can improve the generalization capability of SFS.

3.3 Procedure

With the above described weighting-sample and point injection strategies, the conventional BD based SFS feature selection models are modified as follows.

Step 1. (Initialization) Set the selected feature set *S* empty. Also set the injected point set *D'* empty. Also, for each sample, assign a weight of 1, i.e., $w_i = 1$, $1 \le i \le n$.

Step 2. (Feature selection) From the feature set F, identify the feature f_m which satisfying

$$f_m = \arg\max_{f \in F} \left[BD(f+S)\,|_D + BD(f+S)\,|_{D'} \right].$$

The probability estimators required by BD are established with (7) based on the dataset *D*. And the BDs on *D* and *D'* are defined in (8) and (11) respectively. Put the feature f_m into S and delete it from F at the same time.

Step 3. (Update the sample weights) Set w_i based on equation (6). Then normalize w_i as $w_i = w_i \Big/ \sum_{j=1}^{n} w_j$.

Step 4. (Point-injection) Set *D'* with empty. In the data domain described by *S*, conduct point injection around each sample in the following way.

Around the pattern x_i, produce 10 points based on the distribution $N(x_i, d_i/2)$ where d_i is defined by the equation (10). Place these points into *D'*. Also, the class label and the weight of these injected points are set with y_i and w_i, respectively.

Step 5. If the size in *S* has reached the desired value, then Stop the whole process and output *S*, otherwise Go to step 2.

4 Experimental Results

Our modified SFS, called gradient and point injection based SFS (gp-SFS), is evaluated through comparing with several related methods, namely, the conventional SFS, support machine learning recursive feature elimination scheme (SVM RFE) (Guyon, 2002), and the conventional SFS with the feature-knock-out regularization technique (fko-SFS) (Wolf, 2004). SVM RFE, a typical embedded feature selection model, begins with the training of an SVM (of linear kernel) with all the given features. Then according to the parameters of the trained SVM, features are ranked in terms of importance, and half of the features are eliminated. The training-SVM-eliminating-half-of-features process repeats until no feature is left. The feature-knock-out point injection scheme is designed for classification learning in which a point *x*' is added in each learning iteration. To generate *x*', two samples (say x_1 and x_2) are randomly selected and a feature *f* is specified according to the newly-built model. And all information about *x*' is set with that of x_1, except that $x'(f) = x_2(f)$. We adopt this point injection scheme to modify the conventional SFS as fko-SFS.

To assess the quality of feature selection results, we rely on experimental classification results. In details, given a feature subset for examining, say *S*, certain classifiers are constructed using training data which is also used for feature selection. Then, based on the performance of these classifiers on a test dataset, the quality of *S* is evaluated. Respectable feature subsets should correspond to good classification results. For this evaluation purpose, four typical classifiers are employed. They are multiply percepton model (MLP), support vector machine model with linear kernel (SVM-L), the support vector machine model with RBF kernel (SVM-R) and the 3-NN rule classifier. The MLP used in our study is available at http://www.ncrg.aston.ac.uk/netlab/. For convenience, we set 6 hidden neurons of MLP for all examples. It is worth noting that slightly different number of hidden neurons will not have effect on the overall performance. The number of training cycles is set with 100 in order to ease the concerns on overfitting. And other learning parameters are set with default values. SVM models are available at http://www.isis.ecs.soton.ac.uk/resources/ svminfo.

4.1 Data

Sonar classification. It consists of 208 samples. Each sample is described with 60 features and falls into one of two classes, metal/rock. From 208 samples, 40 ones are randomly selected for training and the others are used for test.

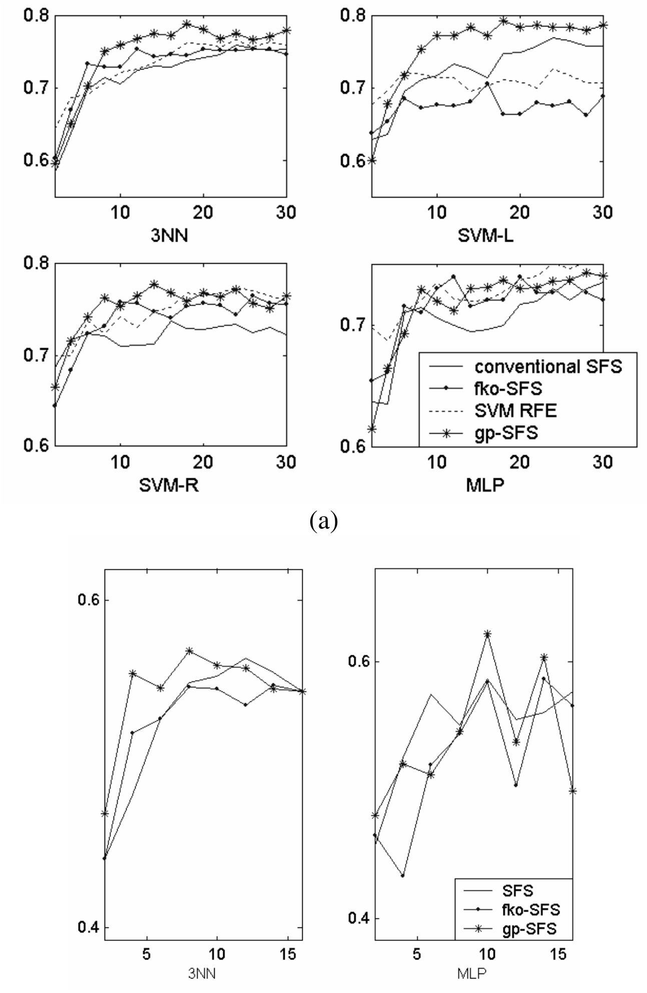
Vehicle classification. This is 4-class dataset for distinguishing the type of vehicle. There are totally 846 samples provided. Each sample is described with 18 features. We randomly select 80 samples for training. The remained 766 samples are used for testing.

Colon tumor classification. This is a microarray data set and is built for colon tumor classification, which contains 62 samples collected from colon-cancer patients (Alon, 1999). Among these samples, 40 samples are tumor, and 22 are labeled "normal". There are 2,000 genes (features) selected based on the confidence in the measured expression levels. We randomly split the 62 samples into two disjoint groups – one group with 31 samples for training and the other one with 31 samples for test.

Prostate cancer classification. This is another microarray dataset, which are collected with the aims to prostate cancer cases from non-cancer cases (Singh, 2002). This dataset consists of 102 samples from the same experimental conditions. And each sample is described by using 12600 genes (features). We split the 102 samples into two disjoint groups – one group with 60 samples for training and the other with 42 samples for testing.

4.2 Results

In each example, we repeat investigation on 10 different sets of training and test data. The presented results are the statistics of 10 different trials. Also, in each training data, the original ratios between different classes are roughly remained. For example,



(a)

(b)

Fig. 1. Comparisons on UCI datasets. (a) sonar classification. (b) vehicle classification.

during the investigation on the colon cancer classification, the original ratio between tumor and normal class, i.e., 40 normal vs. 22 tumor, is roughly kept in each training dataset. For each training dataset, we preprocess it so that each input variable has zero means and unit variance. And the same transformation is then applied to the corresponding test dataset.

The computational complexity of SFS type models is $O(M^2)$ where M is the number of features. A microarray dataset generally contains information of thousands or ten thousands genes. Clearly, directly handling the huge gene sets cost SFSs unbearable computational burden. To improve the computational efficiency, and given by the fact that most genes originally given in a microarray dataset are

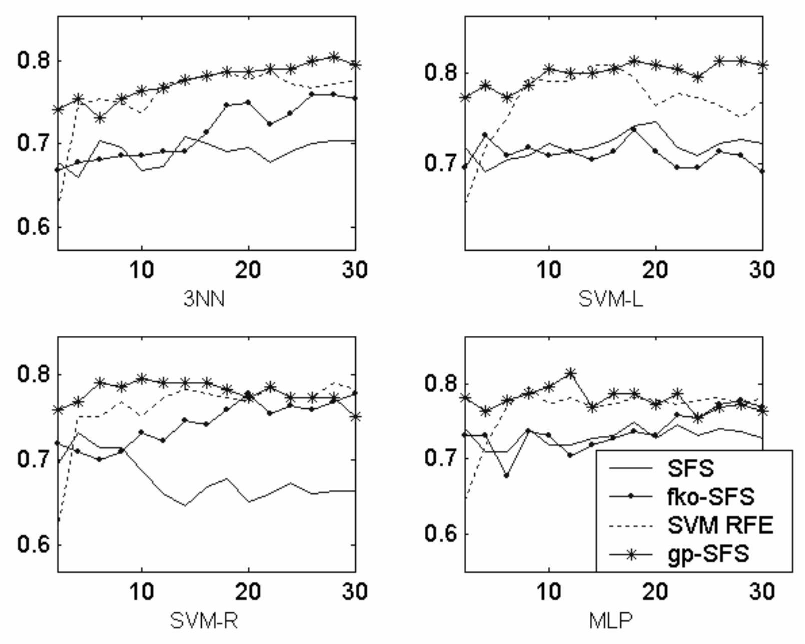


Fig. 2. Comparisons on the colon cancer classification data

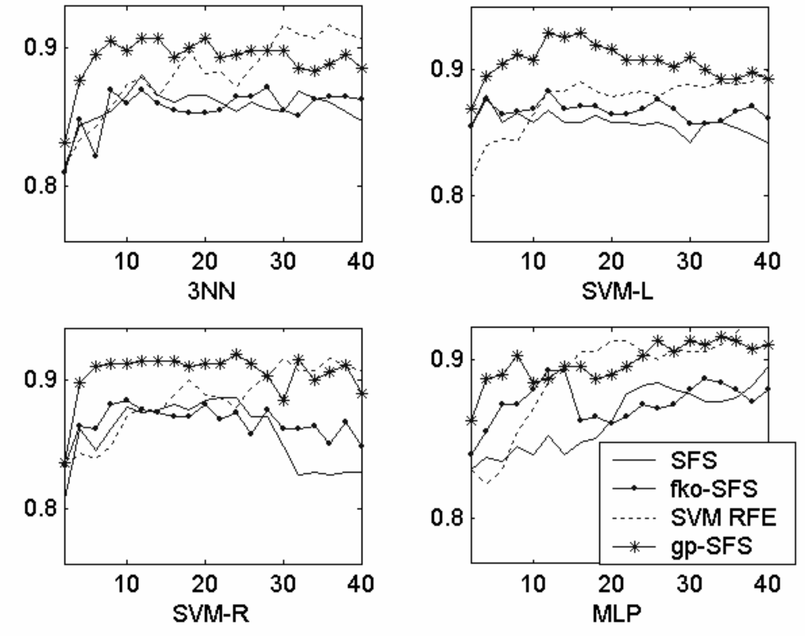


Fig. 3. Comparisons on the prostate cancer classification data

irrelevant to a specified task, a widely used pre-filtering-gene strategy is adopted in our study to eliminate the irrelevant and insignificantly relevant genes before the commencement of feature selection. In details, all the given features (genes) are ranked in a descend order of BD (8). And the one third top-ranked features are left behind for further feature selection.

The comparative results are presented in Figure 1 (for sonar classification and for vehiecle classification), Figure 2 (for colon cancer classification) and Figure 3 (for prostate cancer classification). In most cases, our modified SFS greatly outperform the conventional SFS. This is contributed by the gradient based and point injection strategies. Also, compared with fko-SFS and SVM RFE in which the problem of small-sample is tackled implicitly or explicitly, the proposed SFS still shows its advantages. The contributions of our study can thus be proved.

5 Conclusions

In this paper, two strategies are proposed to enhance the performance of filter feature selection models. The first one is a graident based strategy which is used to enhance the searching effectivenss, and another is a new point-injection approach which is aimed to improve generalization ability. The results obtained on synthetic data and real data obivously demonstrate that these proposed strategies can bring a remarkable improvement. The proposed strategies are only applied to one representative filter model – BD based sequential forward searching. In furture work, we will extend these strategies to other filter models and further evaluate their merits and limitations.

References

Al-Ani A. and Deriche, M. (2000) Optimal feature selection using information maximisation: case of biomedical data, in *Proc. of the 2000 IEEE Signal Processing Society Workshop*, vol. 2, pp. 841-850.

Alon, U. *et al.* (1999) Broad pattern of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. U.S.A.* vol. 96(12), pp. 6745-6750.

Battiti, R. (1994) Using mutual information for selecting features in supervised neural net learning. *IEEE Trans. Neural Networks*, vol. 5, pp537-550.

Bishop, C.M. (1995) *Neural Networks for Pattern Recognition*, New York: Oxford University Press.

Bonnlander, B. (1996) *Nonparametric Selection of Input Variables for Connectionist Learning*, Ph.D. thesis, CU-CS-812-96, University of Colorado at Boulder.

Chow, T. W. S. and Huang, D. (2005) Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information. *IEEE Trans. Neural Networks*, vol. 16, no. 1, pp. 213-224, January 2005.

Devijver, P. A. and Kittler, J. (1982) *Pattern Recognition: a Statistical Approach*, Englewood Cliffs: Prentice Hall.

Golub, T. R. *et al.* (1999) Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286, pp. 531-537.

Glick, N. (1985) Additive estimators for probabilities of correct classification, *Pattern recognition*, vol. 18 (2), pp. 151-159.

Gui, J. and Li, H. (2005) Penalized Cox regression analysis in the high-dimensional and low-sample size settings, with application to microarray gene expression data, *Bioinformatics*, 21(13), pp. 3001-3008.

Guyon, I., Weston J. and Barnhill, S. (2002) Gene selection for cancer classification using support vector machines, *Machine Learning*, vol. 46, pp. 389-422.

Hall, M. A. (1999) *Correlation-based Feature Selection for Machine Learning*, Ph.D. thesis, Department of Computer Science, Waikato University, New Zealand.

Han, J. W. and Kamber, M. (2001) *Data mining: concepts and techniques*. San Francisco: Morgan Kaufmann Publishers, 2001.

Hastie,T., Tibshirani,R. and Friedman,J. (2001) *The Elements of Statistical Learning*, pp. 308-312, Springe.

Huang, D. and Chow, T.W.S. (2005) Efficiently searching the important input variables using Bayesian discriminant*, IEEE Trans. Circuits and Systems*, vol. 52(4), pp.785-793.

Huang, D. , Chow, T.W.S. *et al.* (2005) Efficient selection of salient features from microarray gene expression data for cancer diagnosis, *IEEE Trans. Circuits and Systems, part I*, vol. 52 (9), pp.1909-1918.

Kim, S., Dougherty, E.R, Barrera, J.Y. *et al.* (2002) Strong feature sets from small samples, *Journal of Computational Biology,* 9, pp.127-146.

Lampariello, F. and Sciandrone, M. (2001) Efficient training of RBF neural networks for pattern recognition, *IEEE Trans. On Neural Networks*, vol. 12(5), pp.1235-1242.

Liu, H. and Motoda, H. (1998) *Feature Selection for Knowledge Discovery and Data Mining*, London, GB: Kluwer Academic Publishers.

Matsuoka, S. (1992) Noise injection into inputs in back-propagation learning, *IEEE Trans. Syst., Man, Cybern.,* vol. 22, pp. 436-440.

Molina, L. C., Belanche L. and Nebot, A. (2002) *Feature Selection Algorithms: a Survey and Experimental Evaluation*, available at: http://www.lsi.upc.es/dept/techreps /html/R02-62.html, Technical Report.

Parzen, E. (1962) On the estimation of a probability density function and mode, *Ann. Math. Statistics.,* vol. 33, pp. 1064-1076.

Perkins, S., Lacker K., Theiler, J. (2003) Grafting: Fast, Incremental feature selection by gradient descent in function space, *Journal of machine learning research*, vol. 3, pp. 1333-1356.

Pudil, P., Novovicova, J. and Kittler, J. (1994) Floating search methods in feature selection. *Pattern Recognition Letter*, vol. 15, pp. 1119-1125, Nov. 1994.

Singh, D. *et al.* (2002) Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, vol. 1, pp. 203-209.

Skurichina, M, Raudys, S. and Duin, R.P. (2000) K-nearest neighbours directed noise injection in multilayer perceptron training, *IEEE Trans. On Neural Networks,* vol. 11(2), pp. 504-511.

Wolf, L. and Martin, I. (2004) Regularization through feature knock out, AI memo 2004-2005, available at http://cbcl.mit.edu/cbcl/publications/ai-publications/2004/.

Zagoruiko, N. G., Elkina, V. N. and Temirkaev, V. S. (1976) ZET-an algorithm of filling gaps in experimental data tables. Comput. Syst. vol. 67, pp. 3-28.

Zhou, X., Wang X. and Dougherty, E. (2004) Nonlinear probit gene classification using mutual information and wavelet-based feature selection, *Journal of Biological Systems*, vol. 12 (3), pp. 371-386.

Driven Forward Features Selection: A Comparative Study on Neural Networks

Vincent Lemaire and Raphael Féraud

France Télécom R&D Lannion
vincent.lemaire@orange-ft.com

Abstract. In the field of neural networks, feature selection has been studied for the last ten years and classical as well as original methods have been employed. This paper reviews the efficiency of four approaches to do a driven forward features selection on neural networks . We assess the efficiency of these methods compare to the simple Pearson criterion in case of a regression problem.

1 Introduction

Up to 1997, when a special issue on relevance including several papers on variable and feature selection was published, few domains explored more than 40 features. The situation has changed considerably in the past few years, notably in the field of data-mining with the availability of ever more powerful data warehousing environments. A recent special issue of JMLR [1] gives a large overview of techniques devoted to variable selection and an introduction to variable and feature selection can be found in this special issue [2]. A challenge on feature selection has been organized during the NIPS 2003 conference to share techniques and methods on databases with up to 100000 features. This challenge lead to provide an interesting and exhaustive book [3].

The objective of variable selection is three-fold: improve the prediction performance of the predictors, provide faster and more cost-effective predictors, and allow a better understanding of the underlying process that generated data. Among techniques devoted to variable selection, we find filter methods, which select variables without using a model (for example by ranking them with correlation coefficients), and subset selection methods, which assess subsets of variables according to their usefulness to a given model. Wrapper methods [4] use the elaborated model as a black box to score subsets of variables according to their usefulness for the modeling task. In practice, one needs to define: (i) how to search the space of all possible variable subsets; (ii) how to assess the prediction performance of a model to guide the search and halt it; (iii) how to select the predictor to use.

We discuss in this paper the problem of feature selection and review four methods which have been developed in this field. The main idea is to compare four popular techniques in sense of methods which are integrated in data mining software (Clementine, SAS, Statistica Data Miner...). This paper presents the

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 693–702, 2006.

comparison specifically for neural networks (NN) therefore point (iii) listed above is fixed.

The remainder of the document is organized as follows. Next section deals with classical ingredients which are required in feature selection methods (1) a feature evaluation criterion to compare variable subsets (2) a search procedure, to explore (sub)space of possible variable combinations (3) a stop criterion or a model selection strategy. The section 3 presents the driven forward strategy and four methods to do variable selection with neural networks. Section 4 proceeds with an experimental evaluation on each method on the driven forward strategy for a regression problem.

2 Basic Ingredients of Feature Selection Methods

For all methods in this paper, the notations employed are **(1)** about data distribution: J the number of variables in the full set; I the number of examples in the training set; V_j the variable for which we look for the importance; V_{ij} the realization of the variable V_j for the example i; I_m the input vector part of the example m with n components; $P_{V_j}(u)$ the probability distribution of the variable V_j; $P_I(\nu)$ the probability distribution of examples I; and **(2)** about neural network: OL the output layer; HL the hidden layer; IL the input layer; w_{wz} a weight between a neuron w and a neuron z; f the predictive model (here a neural network); Y_m the output vector part of the example m; and $f_j(a;b) = f_j(a_1, ..., a_n; b) = f(a_1, ..., a_{j-1}, b, a_{j+1}, ..., a_n)$ where a_p is the p^{th} component of the vector a. Finally we note $S(V_j|f)$ as being the importance of the variable V_j using the predictive model f. Note that all methods are presented for an output vector which has only one component but extension to many component is straightforward.

2.1 Features Evaluation

Several evaluation criteria, based either on statistical grounds or heuristics, have been proposed for measuring the importance of a variable subset. For regression, classical candidates are prediction error measures. We will use the mean squared error to compare results in section 4. A survey of classical statistical methods may be found in [5] for regression, [6] for classification, [3] for both; and [7] for neural networks.

2.2 Search Strategy

In general, since evaluation criteria are non monotonous, comparison of feature subsets amounts to a combinatorial problem which rapidly becomes computationally unfeasible. Most algorithms are based upon heuristic performance measures for the evaluation and sub-optimal search. Most sub-optimal search methods follow one of the following sequential search techniques [8]: (a) start with an empty set of variables and add variables to the already selected variable set (forward methods); (b) start with the full set of variables and eliminate variables from the selected variable set (backward methods); (c) start with an empty set and alternate forward and backward steps (stepwise methods). In this paper we will compare criteria only with a driven forward strategy described below.

2.3 Driven Forward Selection

In this paper we define a driven forward selection strategy such as: 1) compute the variable importance using a criterion; 2) rank the variables using the result of the first step; 3) train models where variables are added more and more using the ranking of the variable importance computed in the second step; 4) observe the results versus the number of variables used. This strategy is driven since the first ranking is not questioned and therefore one have at most J model to train.

A simple driven forward strategy uses, for example, the Pearson correlation coefficient which is adapted for linear dependencies[1] and which is not model oriented (it does not take into account the regression model during selection):

$$S(V_j|f) = S(V_j) = \frac{\sum_{i=1}^{I}\left(V_{ij} - \overline{V_j}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_{i=1}^{I}\left(V_{ij} - \overline{V_j}\right)^2 \sum_{i=1}^{I}\left(Y_i - \overline{Y}\right)^2}} \tag{1}$$

For Pearson criterion the driven strategy described is clear since this criterion does not need to use a model in the first step ($S(V_j|f) = S(V_j)$). However any wrapper criterion which allows to measure variable importance could be use in the same way. In this case there is a preliminary step which is to train a model which uses the full set. Then the first step compute the variable importance using this model ($S(V_j|f)$). Others step are not changed. What we can except is that all criteria studied in this paper can achieved better results than using Pearson criterion.

2.4 Stopping Criterion

No stopping criterion has been used in this paper. The performance obtained by each variable selection method has been memorized to be able to plot all results on all selected variables subset with all criteria.

3 Features Selection Methods with Neural Networks Compared

3.1 A Feature Selection Method Based on Empirical Data Probability

The method described here [9] combines the definition of the 'variable importance' as given in Féraud et al. [10] with an extension of Breiman's idea [11].

[1] To capture non linear dependencies, the mutual information is more appropriate but it needs estimates of the marginal and joint densities which are hard to obtain for continuous variables. This method has not been tested in this paper.

This new definition of variable importance both takes into account the probability distribution of the studied variable and the probability distribution of the examples. The importance of an input variable is a function of examples I probability distribution and of the probability distribution of the considered variable (V_j). This method is tested for the first time in this paper on a regression problem.

The importance of the variable V_j is the sum of the measured variation of the predictive model output when examples are perturbed according to the probability distribution of the variable V_j. The perturbed output of the model f, for an example I_i is the model output for this example but having exchanged the j$^{\text{th}}$ component of this example with the j$^{\text{th}}$ component of another example, k. The measured variation, for the example I_i is then the difference between the 'true output' $f_j(I_i;V_{ij})$ and the 'perturbed output' $f_j(I_i;V_{kj})$ of the model. The importance of the variable V_j is computed on both the examples probability distribution and the probability distribution of the variable V_j. The importance of the variable V_j for the model f is then:

$$S(V_j|f) = \iint P_{V_j}(u)duP_I(v)dv\,|f_j\left(I_i;V_{ij}\right) - f_j\left(I_i;V_{kj}\right)| \tag{2}$$

Approximating the distributions by the empirical distributions, the computation of the average of $S(V_j|f)$ would require to use all the possible values of the variable V_j for all examples available such as:

$$S(V_j|f) = \frac{1}{I}\sum_{i\in I}\sum_{k\in I}|f_j\left(I_i;V_{ij}\right) - f_j\left(I_i;V_{kj}\right)| \tag{3}$$

As the variable probability distribution can be approximated using representative examples (P) of an ordered statistic:

$$S(V_j|f) = \frac{1}{I}\sum_{i\in I}\sum_{p\in P}|f_j\left(I_i;V_{ij}\right) - f_j\left(I_i;v_p\right)|\,\mathrm{Prob}(v_p) \tag{4}$$

This method is especially useful when V_j takes only discrete values since the inner sum is exact and not an approximation. View the size of the database used for comparison section 4 P has been fixed to 10 (the deciles are used). For all deciles we chose to used their median as representative values. This approximation allows to speed up the computation and prevents errors which are due to outliers or pathological values.

3.2 A Features Selection Method Based on Neural Networks Weights

This method uses only the network parameter values. Although this is not sound for non linear models, there have been some attempts for using the input weight values in the computation of variable relevance. The weight value in the input layer[2], IL, can provide information about variable importance. The variable importance based on neural networks weights is:

$$S(V_j|f) = \frac{\sum_{z \in HL} \|w_{zj}\|}{\sum_{z \in HL} \sum_{w \in IL} \|w_{zw}\|} \tag{5}$$

3.3 A Features Selection Method Based on Saliency

Several methods propose to evaluate the relevance of a variable by the derivative of the error or of the output with respect to this variable. These evaluation criteria are easy to compute, most of them lead to very similar results. These derivatives measure the local change in the outputs with respect of a given input, the other inputs being fixed. Since these derivatives are not constant as in linear models, they must be averaged over the training set. For these measures to be fully meaningful inputs should be independent and since these measures average local sensitivity values, the training set should be representative of the input space (which is a minimum assumption).

The Saliency Based Pruning method [13] uses as evaluation criterion the variation of the learning error when a variable V_j is replaced by its empirical mean $\overline{V_j}$ (zero if variables are assumed centered). The saliency is:

$$S(V_j|f) = \frac{1}{I}\left(\sum_{i=1}^{I} \left\|f(I_i; V_{ij}) - y_i\right\|^2\right) - \frac{1}{I}\left(\left\|\sum_{i=1}^{I} f(I_i; \overline{V_j}) - y_i\right\|^2\right) \tag{6}$$

This is a direct measure of the usefulness of the variable for computing the output. Changes in MSE are not ambiguous only when inputs are not correlated. Variable relevance being computed once here, this method does not take into account possible correlations between variables.

3.4 A Features Selection Method Based on Output Derivatives

Several authors have proposed to measure the sensitivity of the network transfer function with respect to input V_j by computing the mean value of outputs derivative with respect to V_j over the whole training set. Most measures use average squared or absolute derivatives [14,15,16]. The variable importance is: $S(V_j|f) = \frac{1}{I}\sum_{i=1}^{I}\left(\partial f / \partial V_j(V_{ij})\right)$. These measures being very sensitive to the input space representativeness of the sample set, several authors have proposed to use a subset of the sample in order to increase the significance of their relevance measure. In order to obtain robust methods, "non-pathological" training examples should be discarded. A parameter, here ϵ, is needed to adjust the range variation over V_j given an example (V_{ij}). In this paper we choose to use the definition:

$$S(V_j|f) = \frac{1}{I}\sum_{i=1}^{I} |f_j(I_i, Vij - \epsilon) - f_j(I_i, Vij + \epsilon)| \tag{7}$$

[2] A more sophisticated heuristic, but very close to the one above in case of a single output neuron, has been proposed by Yacoub and Bennani [12], it exploits both the weight values and the network structure of a multilayer perceptron.

4 Experimental Results on Orange Juice Database

4.1 Experimental Conditions and Results Presentations

Database: The database has been provided by Prof. Marc Meurens, Université Catholique de Louvain, BNUT unit. The goal is to estimate the level of saccharose of an orange juice from its observed near-infrared spectrum. The training set is constituted of 150 examples described by 700 features (variables) and the test set is constituted of 68 examples described also by 700 features. There is no missing value and variables are continuous but note that the number of training examples (150) is more of four times as small as the number of features (700). Nothing else is known about this database (see http://www.ucl.ac.be/mlg/index.php?page=DataBases). The preprocessing used for input variable as well as for output variable is only a min-max standardization. All the results presented below (the mean squared error) are computed on the standardized output.

Cross Validation: For all experimental conditions, 25 trainings are performed with different initialization of the weights and different training, validation set as follow: we have drawn a training set (100 examples) from the training set available on the web site (among 150) and the others example of the training set has been used as a validation set. Each training is stopped when the cost (the mean squared error) on the validation set does not decrease since 200 iterations. At the end of each training, the global mean squared error on the test set is computed for comparison purposes. In the driven forward strategy the variables importance are not questioned. So, when one gives results over 25 training there are results over 25 forward procedures (for a given step, a given number of variables, the variables chosen are not necessary the same to compute the mean errors presented in Figure 7).

Neural network topology and training parameters: A single multilayer perceptron with 1 hidden layer, tangent hyperbolic activation function and stochastic back-propagation of the squared error as training algorithm has been used. Using full set of variables the learning rate has been determined to be α=0.001 and the number of hidden unit has been determined to be HL=15. Again, these parameters has been evaluated over 25 training from a range variation of α from 0.0001 up to 0.1 and HL from 1 up to 30.

Regularization: The orange juice database is constituted of 700 variables which are very correlated to the output target (see Figure 1, coefficients between normalized input variables and the normalized output). Methods presented above test the importance of all variables one by one so a successful regularization method has to be employed. We added a regularization term active only on directions in weight space which are orthogonal to the training update [17]. This regularization prevents correlation effects between input variables without learning degradations. The regularization term (in batch procedure for it) has been always 10^{-3} of the learning rate.

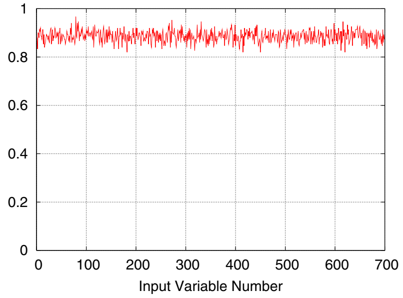


Fig. 1. Absolute Pearson coefficient

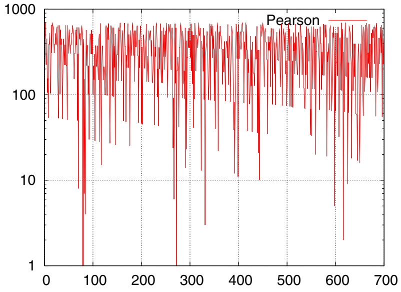


Fig. 2. Ranking of Pearson coefficient

4.2 Comparison Using the Full Set and a Same Neural Network

Figures 3,4,5,6 show variable importance found using the five criterion described above (except Pearson criterion for which one can see this representation in Figure 2) and computed with the same neural networks trained with the full set of variables. Figure 3, Figure 4, Figure 5, Figure 6 show respectively versus the number of the variables the "Norm Importance" obtained using equation 5, "Saliency Importance" obtained using equation 6, "Local Importance" obtained using equation 7 and "Global Importance" obtained using equation 4 . On all sub figure horizontal axis represents the number of the variables and vertical axis represents (in log-scale to focus on first important variables) the ranking of the variables from 1 (the most useful) to 700 (the less useful). This representation identifies clearly first important variables for all criterion using the same neural network and allows to compare behaviors.

The four criteria Norm, Saliency, Local and Global do not agree with Pearson criterion (see Figure 2). For criteria Norm, Local and Global important variables are near the six hundredth variable. Saliency criterion selects variables near the 130th. Global criterion ranks this group after the group near the six hundredth variable. Among group near the 600th variable Global criterion does not order variables as Norm and Local criteria (the 562th before the 592th). Norm and Local criteria very agree on this regression problem. Results presented in next section with the driven forward procedure will give more results elements.

4.3 Results with the Driven Forward Strategy

Whatever is the neural network trained the results obtained using Pearson criterion will be the same since this criterion does not use the model to compute variable importance. But it is not the case for others criteria described above. The ranking obtained can depend on the neural network trained and therefore of its initialization, the order to present examples, etc... For all criteria 20 neural networks ($k = 20$) have been trained using the full set of variables. The mean value of the criterion has been computed on all neural networks such as: $\overline{S(V_j|f)} = 1/k \sum_k S(V_j|f_k)$. Using this mean value on all variables a ranking has been determined. Table 1 presents this ranking. Then this ranking has not been

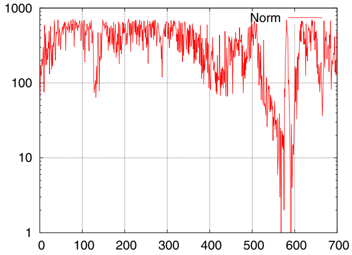


Fig. 3. 'Norm Importance'

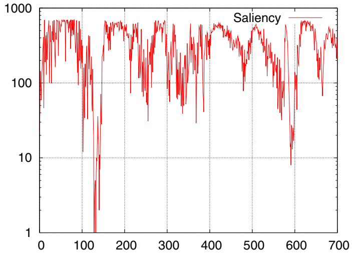


Fig. 4. 'Saliency Importance'

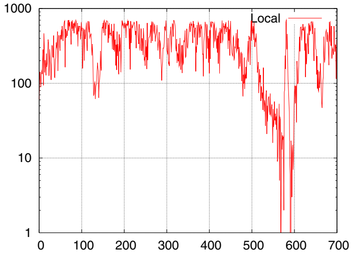


Fig. 5. 'Local Importance'

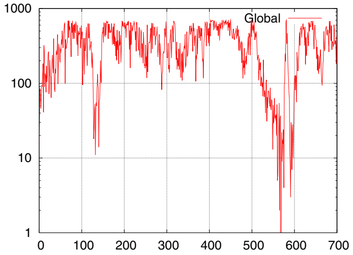


Fig. 6. 'Global Importance'

questioned. It is used to train neural networks which used one, two or more important variables. Experimentations have been made twenty times to obtained mean results using one, two or more important variables on all criteria.

Table 1. The ten more important variables

Pearson	80	273	85	332	617	71	83	268	599	118
Norm	595	596	592	593	590	594	591	570	597	598
Saliency	595	131	1	2	129	3	130	592	6	593
Local	595	592	596	593	590	594	591	599	597	598
Global	570	595	592	596	590	593	594	572	569	571

The Figure 7 presents results obtained with the four methods and Pearson criterion which is a baseline results. Results after 100 variables are not presented since they are the same for all criteria and are the same than using the full set. Each plot represents the mean results of the mean squared error on the normalized output through 20 forward procedures. The standard deviation is not represented for reading reasons and a figure which is not overloaded. The standard deviation is ± 0.003 for all points. For example, for the Pearson criterion and using ten variables, the result is therefore 0.045 ± 0.003.

On this regression problem, which is compose of a full set of 700 variables and few examples for training, we observe the following ranking of criteria (from the best to the least): (1) Global, (2) Saliency (3) Local and Norm, (4) Pearson. With less than 100 variables criteria Global, Saliency and Pearson obtain the

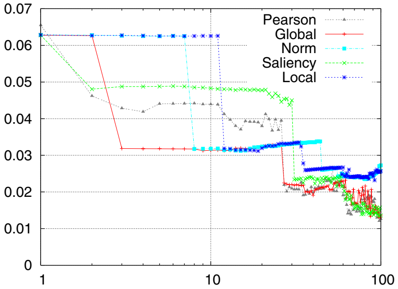


Fig. 7. Mean squared error using driven forward strategy versus the number of variables used

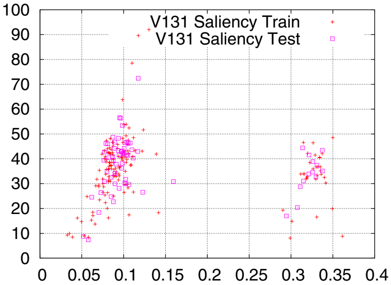


Fig. 8. Neural network outputs versus the ordered values of the 131th variable

same results than using all variables (0.011 ± 0.003). To obtain this performance Norm and Local criteria need 150 variables. Significant degradations on results appear under 60 variables on all criteria. The Global criterion gives excellent and best results: better performances of the neural network trained are always obtained before others (until all criteria allow to obtained same results).

To analysis more in depth the difference in term of performances we focus on the 131th variable since there is a disagreement between criteria for this variable. We plot on Figure 8 ordered values of the 131th variable on horizontal axis and the estimated output on the vertical axis (using the same neural network as in section 4.2). Clearly for this variable, which constituted by two groups of values, it is not relevant to measure its importance with saliency: its mean is out of the data distribution. This discontinuity explains the overestimation of the variable importance using Saliency criterion. On the other hand, Local criterion does not rank this 131th variable in the ten most important variables since derivatives importance is not adapted to bimodal distribution. The Global criterion where data distribution is used is able to take into account bimodal distribution. It ranks this variable as an important variable. This type of difference in behaviors explains the difference in performances.

5 Conclusion

These comparisons show that, on this real application, it is possible to obtain excellent performances with the four criteria with a large preference for the Global criterion; knowing that the database used is a particular database with very correlated variables and few examples compare to the number of the full set of variables. Future work should address experiments on larger data sets[3].

[3] as for example `http://theoval.cmp.uea.ac.uk/~gcc/competition/`

References

1. JMLR, ed.: Special Issue on Variable and Feature Selection. Volume 3(Mar). Journal of Machine Learning Research (2003)
2. Guyon, I., Elisseef, A.: An introduction to variable and feature selection. JMLR **3(Mar)** (2003) 1157–1182
3. Guyon, I.: To appear - Feature extraction, foundations and applications. - (2006)
4. Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intelligence **97(1-2)** (1997)
5. Thomson, M.L.: Selection of variables in multiple regression part i: A review and evaluation and part ii: Chosen procedures, computations and examples. International Statistical Review **46:1-19 and 46:129-146** (1978)
6. McLachlan, G.: Discriminant Analysis and Statistical Pattern Recognition. Wiley-Interscience publication (1992)
7. Leray, P., Gallinari, P.: Feature selection with neural networks. Technical report, LIP6 (1998)
8. Miller, A.J.: Subset Selection in Regression. Chapman and Hall (1990)
9. Lemaire, V., Clérot, C.: An input variable importance definition based on empirical data probability and its use in variable selection. In: International Joint Conference on Neural Networks IJCNN. (2004)
10. Féraud, R., Clérot, F.: A methodology to explain neural network classification. Neural Networks **15** (2002) 237–246
11. Breiman, L.: Random forest. Machine Learning **45** (2001)
12. Yacoub, M., Bennani, Y.: Hvs: A heuristic for variable selection in multilayer artificial neural network classifier. In: ANNIE. (1997) 527–532
13. Moody, J.: Prediction Risk and Architecture Selection for Neural Networks. From Statistics to Neural Networks-Theory and Pattern Recognition. Springer-Verlag (1994)
14. Ruck, D.W., Rogers, S.K., Kabrisky, M.: Feature selection using a multilayer perceptron. J. Neural Network Comput. **2**(2) (1990) 40–48
15. Réfénes, A.N., Zapranis, A., Utans, J.: Stock performance using neural networks: A comparative study with regression models. Neural Network **7** (1994) 375–388
16. Refenes, A., Zapranis, A., Utans, J.: Neural model identification, variable selection and model adequacy. In: Neural Networks in Financial Engineering, Proceedings of NnCM-96. (1996)
17. Burkitt, A.N.: Refined pruning techniques for feed-forward neural networks. Complex System **6** (1992) 479–494

Non-negative Matrix Factorization Based Text Mining: Feature Extraction and Classification

P.C. Barman, Nadeem Iqbal, and Soo-Young Lee*

Brain Science Research Center and Computational NeuroSystems Lab,
Department of BioSystems,
Korea Advanced Institute of Science and Technology
Daejeon, 305-701, Republic of Korea
{pcbarman, nadeem}@neuron.kaist.ac.kr,
sylee@neuron.kaist.ac.kr

Abstract. The unlabeled document or text collections are becoming larger and larger which is common and obvious; mining such data sets are a challenging task. Using the simple word-document frequency matrix as feature space the mining process is becoming more complex. The text documents are often represented as high dimensional about few thousand sparse vectors with sparsity about 95 to 99% which significantly affects the efficiency and the results of the mining process. In this paper, we propose the two-stage Non-negative Matrix Factorization (NMF): in the first stage we tried to extract the uncorrelated basis probabilistic document feature vectors by significantly reducing the dimension of the feature vectors of the word-document frequency from few thousand to few hundred, and in the second stage for clustering or classification. In our propose approach it has been observed that the clustering or classification performance with more than 98.5% accuracy. The dimension reduction and classification performance has observed for the Classic3 dataset.

1 Introduction

Text Mining is the process of discovering useful knowledge or patterns from unstructured or semi-structured text. The clustering or categorizing of text documents are one of the fundamental part of the text mining process. One of the great challenges for today's information science and technology is to develop algorithms and software for efficiently and effectively organizing, accessing and mining the information from a huge amount of text collection.

Feature extraction of the huge collection of textual data is important factor to achieve an efficient and effective algorithm for categorize the unstructured text data. Many researchers have given their attention to reduce the dimension of the document feature vector. In this paper, we focus on the task of reducing the document feature vector and classifying text documents into a pre-defined set of topical categories, commonly referred to as document clustering which is an enabling technology for information processing applications.

[*] To whom it will be correspondent.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 703 – 712, 2006.

The NMF algorithm has been using successfully for semantic analysis [1]. NMF algorithm shows an outperform in document clustering [3] over the methods such as singular value decomposition and is comparable to graph partitioning methods, K-mean clustering [4], probabilistic clustering using the Naive Bayes [5] or Gaussian mixture model [6] etc. F Shahnaz et al. [7] cluster the text documents by imposing sparsity constrain into the NMF algorithm this sparsity constrains makes slow convergence of the algorithm.

Another related line of research is the simultaneous clustering approach I. S. Dhillon [8] information theoretic co-clustering of join probability distribution of two random variables or co-clustering, or bipartite graph partitioning Zha et al., [9] to reduce the dimensionality of feature vectors. Jia Li et al [10] use the two-way Poisson mixture models to reduce the dimension of the document feature vectors. One common approach has associated with these methods is that they all consider the whole document collection which gives a very high dimensional document feature vector at starting point.

The general paradigm i.e., term-frequency document matrix of representing text documents are more commonly using approach. The elements of the matrix V = [v_{ij}] where v_{ij} is the term frequency i.e., the number of times word i occurs in document j. Each document is represented as a collection of an n-dimensional vector. The number of distinct words in any single document is usually smaller than the size of the vocabulary, leading to sparse document vectors, vectors with many zero components which make the classification algorithms more challenging.

In our approach, we reduce the sparsity of the document vectors by reducing the number of insignificant words as a result of decreasing the correlation coefficient among the feature vectors, which increase the classification performance. We used two stage NMF algorithms: in the first stage we reduce the feature dimension for each document vectors, and the second stage we used for clustering or classification of the text documents. We explain the approaches details in section 3.

2 Nonnegative Matrix Factorization (NMF) Algorithm

Given a non-negative n x m matrix V; find non-negative factors, W, of n x r matrix, and H, r x m, such that: $V \approx WH$ or

$$V_{ij} \approx (WH)_{ij} = \sum\nolimits_a W_{ia} H_{aj} \tag{1}$$

where r is chosen as $r < nm/(n+m)$

V is the word-frequency matrix; W is basis feature matrix; H is encoded matrix and it is one-to-one correspondence with a sample of V.

For our application purpose we make a single modification in the update rule of [1] [2]. In our case we also normalize the encoding matrix H, like W, which are as follows,

$$H_{kj}^{(n+1)} = H_{kj}^{(n)} Q_D \left(\left[W^{T(n)} \right]_{ki}, \frac{V_{ij}}{\left[W^{(n)} H^{(n)} \right]_{kj}} \right)_{kj} \tag{2}$$

$$H_{kj}^{(n+1)} = \frac{H_{kj}^{(n+1)}}{\sum_k H_{kj}^{(n+1)}} \tag{3}$$

$$W_{ik}^{(n+1)} = W_{ik}^{(n)} Q_D \left(\frac{V_{ij}}{\left[W^{(n)} H^{(n)}\right]_{ij}}, \left[H^{T(n+1)}\right]_{kj} \right)_{ik} \tag{4}$$

$$W_{ik}^{(n+1)} = \frac{W_{ik}^{(n+1)}}{\sum_i W_{ik}^{(n+1)}} \tag{5}$$

Where $Q_D(A,B)_{ij} = \sum_k A_{ik} B_{kj} = AB$, and all the $(.)_{ij}$ indicates that the noted division and multiplications are computed element by element.

3 Propose NMF Model and Data Set

We propose the two-stage NMF model in order to reduce the feature vector dimension and reduce the complexity of the clustering or classification process of the text data.

The first stage is for feature extraction of the text data using the basis-probability model of the basis vectors obtained by the NMF algorithm discuss in section 3.2 and in the second stage we used the NMF algorithm for classification of the text documents. We normalize the encoding matrix H to find the relevant probability of the documents to a certain cluster or category. Typically, the basis vectors W_i are random probability distribution of high dimension. We approximate these vectors as exponential probability distribution and we defined these distributions as *basis-probability distributions*. The overall model of our works shows in *figure1*.

For our experiment we have consider the Classic3[1] text data set. This corpus consists of 3891 abstract of three different journal articles. The distribution of the articles is as follows: MEDLINE: 1033 abstracts from medical journal, CISI: 1460 abstracts from information retrieval journal, CRANFIELD: 1398 abstracts from aeronautical systems papers.

3.1 Pre-processing

In the pre-processing step we have randomly selected about 15% representative or training documents from the whole document collection. We have filtered out some English stop words[2] such as 'the', 'to', the numerical values, and the special characters such as '<', '=', etc. After removing the words or characters, find the term-frequency document matrix V. Let $\mathrm{T} = \{ \mathrm{t}_1, \mathrm{t}_2, \ldots\ldots, \mathrm{t}_\mathrm{n} \}'$ be the complete vocabulary set of the training documents where t_i is the ith word or term in the vocabulary set. The term-document frequency vector for document i is $v_i = \{x_{1i}, x_{2i}, \ldots\ldots, x_{ni}\}$ where x_{ji} represents the frequency of the term j in document i.

[1] http://www.cs.utk.edu/~list

[2] http://www.perseus.tufts.edu/Texts/engstop.html

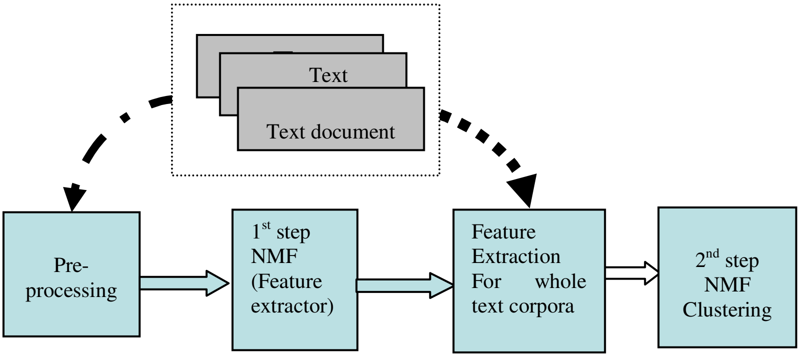

Text
Text document
Pre-processing
1st step NMF (Feature extractor)
Feature Extraction For whole text corpora
2nd step NMF Clustering

Fig. 1. Text Document clustering process using NMF algorithms

3.2 Create a Vocabulary Set of Significant Terms

In the first step NMF algorithm of figure 1, using update rules equations 2 to 5 of NMF algorithm we factorize the term-frequency document matrix V into non-negative basis matrix W and encoding matrix H. The basis feature vectors Wi, i= 1, 2,…, r, represents is the number of clusters or categories of the text data are random probability distribution of high dimension. To select r we measure the correlation coefficient R among the basis vectors W_i for different values of r using equation (6) than find *min(max(* R *))* which means to find the maximum independency among the basis feature vectors the results shown in table 1.

$$R_{ij} = \frac{\sum_{k=1}^{n} (w_{ki} - \overline{w}_i)(w_{kj} - \overline{w}_j)}{\sqrt{\sum_{k=1}^{n} (w_{ki} - \overline{w}_i)^2 \sum_{k=1}^{n} (w_{kj} - \overline{w}_j)^2}} \tag{6}$$

where n = basis feature vector dimension and {i # j} = 1, 2,…,r. We consider the basis feature as basis-probability distribution and arrange the terms of each basis vector in descending order according to probability. The probability of each vector decreases exponential with increasing the number of terms. Now convert the basis-probability distribution into logarithmic scale as shown in figure 2 and truncate the nonlinear portions of the feature vectors. As a result we get the feature vectors of significant terms with reduced dimension. We make a new vocabulary set T_{new} by considering the terms of each reduced feature vectors.

3.3 Feature Extraction for Whole Text Corpora

Using the vocabulary set T^{new} and let $|T^{new}|$ = N, we extract the word-document frequency feature matrix V^{new} for the whole corpus to classify the documents or any new coming documents which is relevant to this document collection. Now the term-frequency vector for ith document is defined as $v_i^{new} = [x_{1i}, x_{2i}, \ldots, x_{Ni}]^T$ where x_{ji} represents the frequency of the term j in document i, in this case: i=1, 2 …m, total number of documents, and j=1, to N (the size of new vocabulary set).

3.4 NMF Clustering

The second stage NMF algorithm of figure 2 factorizes the new word-frequency matrix V^{new} into two factors W (the basis weight matrix) and H (the encoding matrix) using same update rules as before. The encoding matrix H of dimension r×m, (where r is the number of clusters and m is the numbers of documents) has been used to cluster the documents. Since the matrix H is a column wise normalized as in equation (3) it represents the relevant probability of the documents corresponding to each row i.e., to each cluster. In our case we consider the maximum probability for clustering the documents. Let us consider a particular document j its i^{th} row has maximum then consider the j^{th} document in class or cluster i.

4 Experiments

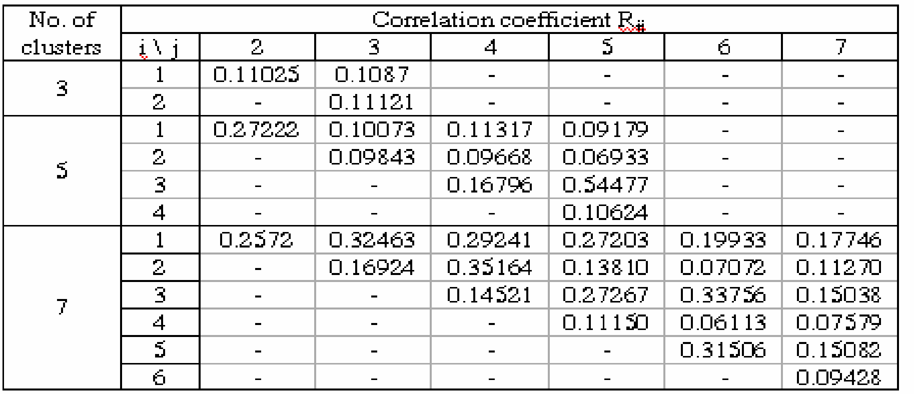
At first we try to find the number of distinct classes or clusters in the given text data base, to do this we randomly chose about 15% (600 out of 3891) documents of the total number of text collection. After preprocessing and making the word-frequency matrix we apply the NMF algorithm in this case the vocabulary size is n=7972, number of documents m=600. Initially we consider r = 3, 5 and 7 then calculate the correlation coefficient R equation (6). Finally choose r=3 because for r = 3, we get minimum value for maximum (R) as shown in table 1.

After fixing the number of clusters (in our case r = 3) we extract the basis feature vectors and plot in the logarithmic scale as shown in the figure 2. We consider the terms only in the approximate linear region of the curve. To find the appropriate numbers of significant terms for better clustering or classification performance we consider different number of terms. Figure 3 represents the reduced basis feature vectors (only for four cases with dimension 300, 500, 700, and 900) in the logarithmic scale. With in the linear region of the curve we consider the feature vector dimension from 50 to 1500 terms and tested the correlation coefficient among the feature vectors as shown in table 2 and figure 4 to check the clustering performance. We test the clustering or classification efficiency by considering different values of feature vectors dimension.

5 Results

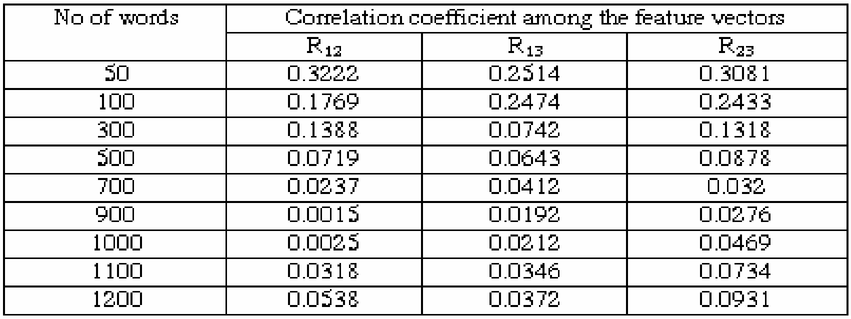
It has been observed that for r=3, the value of $min(max(\text{R}))$ is minimum which means the basis vectors show maximum independency among them as a result we consider there are 3 clusters in the text collection. We consider the threshold of maximum independency is less than 15%. For example in the case of r=5, the basis vectors 3 and 5 are mostly correlated each other since the correlation coefficient among them is 0.54477 similarly for the case of r= 7. We have also tried to reduce the correlation among the feature vectors by excluding some common terms or words from the basis vectors.

Table 1. Correlation coefficient among the basis vectors for various number of clustering r



No. of clusters	Correlation coefficient R_{ij}						
	i \ j	2	3	4	5	6	7
3	1	0.11025	0.1087	-	-	-	-
	2	-	0.11121	-	-	-	-
5	1	0.27222	0.10073	0.11317	0.09179	-	-
	2	-	0.09843	0.09668	0.06933	-	-
	3	-	-	0.16796	0.54477	-	-
	4	-	-	-	0.10624	-	-
7	1	0.2572	0.32463	0.29241	0.27203	0.19933	0.17746
	2	-	0.16924	0.35164	0.13810	0.07072	0.11270
	3	-	-	0.14521	0.27267	0.33756	0.15038
	4	-	-	-	0.11150	0.06113	0.07579
	5	-	-	-	-	0.31506	0.15082
	6	-	-	-	-	-	0.09428

Table 2. Absolute value of correlation coefficient among the basis vectors for various number of words for three clusters



No of words	Correlation coefficient among the feature vectors		
	R_{12}	R_{13}	R_{23}
50	0.3222	0.2514	0.3081
100	0.1769	0.2474	0.2433
300	0.1388	0.0742	0.1318
500	0.0719	0.0643	0.0878
700	0.0237	0.0412	0.032
900	0.0015	0.0192	0.0276
1000	0.0025	0.0212	0.0469
1100	0.0318	0.0346	0.0734
1200	0.0538	0.0372	0.0931

Figure 2 represents the logarithmic probability of the basis vectors. In our experiment, at first we arrange the basis vectors in descending order according to each term probability to the corresponding vectors. We observed that due to the sparsity of the basis vectors the log of probability become abruptly low after certain number of terms (about 1200). We choose the terms within the linear part of log probability distribution. In this approach we try to achieve the maximum limit of the significant terms in each basis vector. Figure 3 represents the linearly decreasing the log probability of the feature vectors for various number of words or terms.

Table 2 represents the observed the absolute value of the correlation coefficient among the basis vectors. Out intension is to achieve the basis vectors as independent as possible. It has been observed that the absolute value of correlation coefficients are minimum near zero for a certain number of indices and increase either decreasing or increasing the number of terms as shown in figure 4. We have got better classification

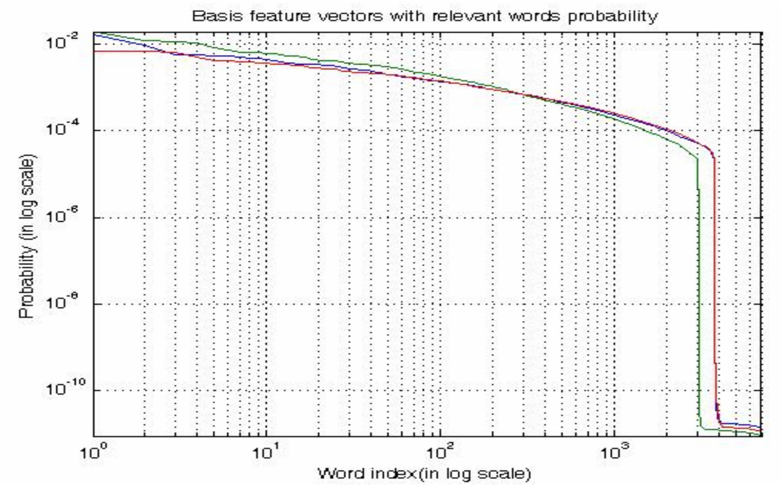


Fig. 2. This figure represents the probabilities of words relevant to each basis feature vectors. The X-axis represents the number of distinct word's index which is equal to the initial vocabulary size (7972); Y-axis is the relative probability or strength of the words corresponding to the feature vectors.

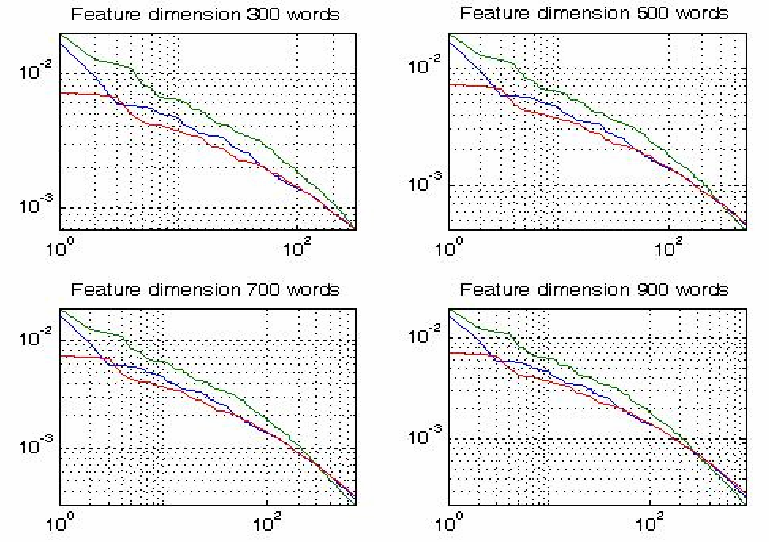


Fig. 3. This figure represents the probabilities of words relevant to each basis feature vectors with different dimensions (300, 500, 700, and 900 words). The X-axis represents the number of distinct word's index and Y-axis is the relative probability or strength of the words corresponding to the feature vectors.

performance of the NMF algorithm within the feature dimension from 200 to 1200 words. For very low dimension of the feature vectors the features are insufficient for proper classification and for high dimension the basis vectors are become correlated so reduce the performance as shown in figure 5. Table 3 represents the clustering or classification performance of the NMF algorithm, first three columns before reducing the feature vectors dimension for the training case, next three columns after reducing the dimension it has been observed that the clustering performance increased. The last three columns show the classification performance for whole documents, the result is well comparable with Inderjit S. Dhillon et. al. [8] reported in SIGKDD '03.

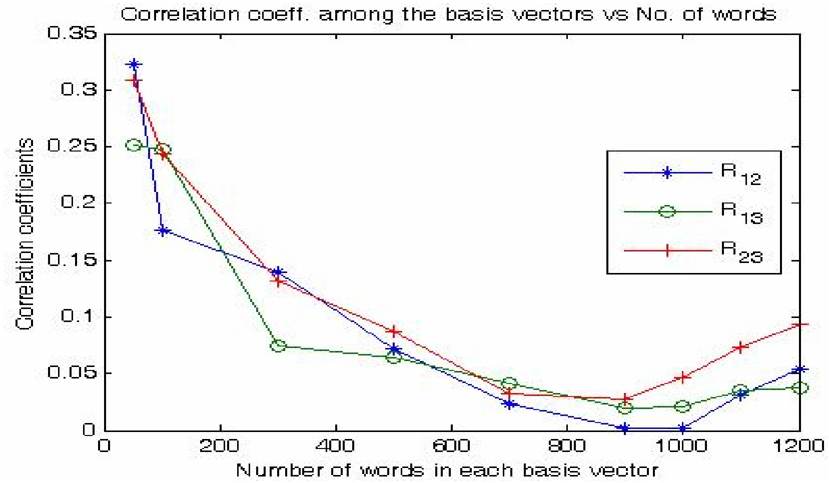
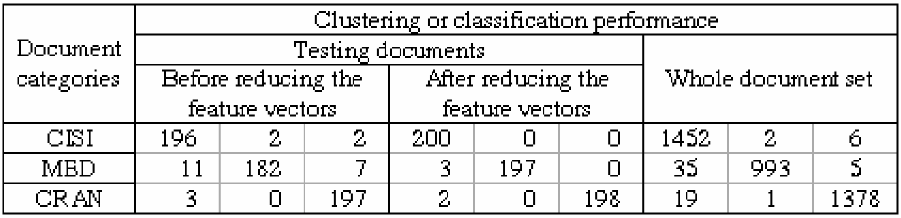


Fig. 4. This figure represents the correlation coefficients among the feature vectors corresponding to the number of words in each basis feature vectors. The X-axis represents the number of words in the feature vectors and Y-axis represents the relative correlation coefficient among the feature vectors.

Table 3. The clustering performance of the NMF algorithm where the first two column show the performance for the training set of 600 documents (200 from each categories) and the last column shows for whole Classic3 data set



Document categories	Clustering or classification performance								
	Testing documents						Whole document set		
	Before reducing the feature vectors			After reducing the feature vectors					
CISI	196	2	2	200	0	0	1452	2	6
MED	11	182	7	3	197	0	35	993	5
CRAN	3	0	197	2	0	198	19	1	1378

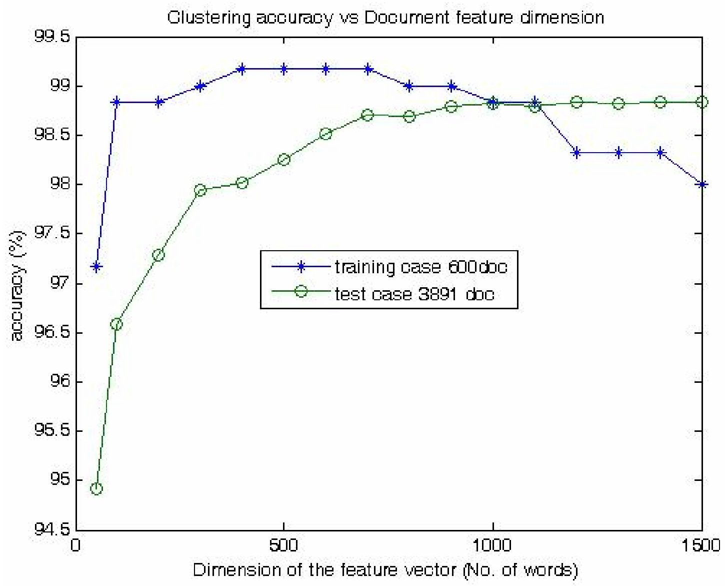


Fig. 5. This figure represents the classification performance for training data set (600 documents) and whole document set (3891 documents). X-axis represents the feature dimension, and Y-axis represents the accuracy.

6 Conclusion and Future Works

The main focus point in our work is make the feature vectors are independent i.e., to reduce the correlation coefficient among the feature vectors near to zero by reducing the dimension or vocabulary size. As we know the feature of the text data is very sparse means the major portion of the word-frequency matrix is zero and it is mainly depends on the number of the unique words in the vocabulary set. There so many non-significant words which don't have major contribution for clustering or categorizing the documents. NMF is a very simple and effective algorithm to reduce the dimension of the feature vectors of the text data. It is also simple and adaptive algorithm for document clustering. By reducing the feature vectors dimension in the training stage of the document clustering process it significantly helps to save the learning time and memory, and also increase the clustering efficiency as we have seen the sparsity of the document feature reduce the clustering or classification efficiency.

In this document we have present only the hard-clustering. In future we will try to soft-clustering. We will also try to sub-clustering (tree like clustering) the documents. We will also try the text-base user identification for an intelligent office-assistant system.

Acknowledgement

This research was supported as the Brain Neuroinformatic Research Program by Korean Ministry of Commerce, Industry, and Energy.

References

1. D. D. Lee and H. S. Seung: Learning the parts of objects by non-negative matrix factorization. Nature, 401(1999):788–791.
2. D. D. Lee and H. S. Seung: Algorithms for non-negative matrix factorization. In Advances in Neural Information Processing 13 (Proc. NIPS*2000), MIT Press, 2001.
3. W. Xu, X. Liu, and Y. Gong.: Document-Clustering based on Non-Negative Matrix Factorization. In Proceedings of SIGIR'03, July 28-August 1, pages 267–273, Toronto, CA, (2003).
4. P. Willett.: Document clustering using an inverted file approach. Journal of Information Science, 2:223–231, (1990).
5. L. Baker and A. McCallum: Distributional clustering of words for text classification, In Proceedings of ACM SIGIR, (1998).
6. X. Liu and Y. Gong.: Document clustering with cluster refinement and model selection capabilities, In Proceedings of ACM SIGIR 2002, Tampere, Finland, (2002).
7. Farial Shahnaz and Michael W. Berry; Document Clustering Using Nonnegative Matrix Factorization; Journal on Information Processing & Management; Elsevier (2004).
8. Inderjit S. Dhillon,: Subramanyam Mallela, Dharmendra S. Modha; Information-Theoretic Co-clustering, SIGKDD '03, August 24-27, 2003, Washington, DC, USA (2003).
9. Zha, H., He, X., Ding, C., Gu, M., Simon, H.,: Bipartite graph partitioning and data clustering, In Proceedings of ACM CIKM. (2001).
10. Jia Lia, Hongyuan Zha: Two-way Poisson mixture models for simultaneous document classification and word clustering, Computational Statistics & Data Analysis, Elsevier (2004)

Adaptive Parameters Determination Method of Pulse Coupled Neural Network Based on Water Valley Area

Min Li[1,2], Wei Cai[2], and Zheng Tan[1]

[1] School of Electronics and Information Engineering, Xi'an Jiaotong University,
Xi'an 710049, Shaanxi Province, P.R. China
limin@mailst.xjtu.edu.cn
[2] Xi'an Research Inst. of Hi-Tech, 710025, Shaanxi Province, P.R.C.

Abstract. Pulse coupled neural network (PCNN) is different from traditional artificial neural networks, models of which have biological background and are based on the experimental observations of synchronous pulse bursts in the cat visual cortex. However, it is very difficult to determine the exact relationship between the parameters of PCNN model. Focusing on the famous difficult problem of PCNN, how to determine the optimum parameters automatically, this paper proposes the definition of water valley area, establishes a modified PCNN, and puts forward an adaptive PCNN parameters determination algorithm based on water valley area. Extensive experimental results on image processing demonstrate its validity and robustness.

1 Introduction

Pulse-coupled neural network (PCNN) based on Eckhorn's model of the cat visual cortex has great significant advantage in image processing, including segmentation, target recognition *et al*[1,2]. However, the performance depends on the suitable PCNN parameters, which are tuned by trial so far.

During recent years, some work on determining the optimal values of PCNN parameters has been done. Some of them are concentrated on optimizing single parameter while keeping others fixed [3,4,5]. Some train the parameters with desired images to achieve the optimal values [6].

G. Kuntimad and H. S. Ranganath [2] have provided conditions for perfect image segmentation using PCNN. However, the conditions and algorithm are only fit for those applications with single object and single background, which is too strict to be applied to usual images.

Ma Y.D. *et al.* [3] have proposed a new PCNN algorithm of automatically determining the optimum iteration times *N* based on the entropy of segmented image. It is the criterion of maximal entropy of segmented binary image of PCNN output. Lots of experiments based on this method showed that images can be segmented well when the pixel numbers of object and background are nearly the same. But when the pixel numbers of object and background are different significantly, the segmentation performs badly.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 713 – 720, 2006.

Liu, Q., *et al.* [4] have proposed an improved method based on reference [2], in which cross-entropy is put forward to replace maximal Shannon entropy as the criterion of cyclic iterations times *N*. However, the segmented results are lack of adaptability just as the approach in reference [3].

Currently, adopting simplified PCNN model to decrease the parameters number is an important trend in image segmentation field. There are lots of simplified PCNN models [5~11].

Karvonen, J.A. [6] has presented a method for segmentation and classification of Baltic Sea ice synthetic aperture radar (SAR) images, based on PCNN. As the authors mentioned, a very large set of data representing different sea ice conditions should be required to optimize PCNN parameters, which is unfeasible in most applications.

Since image segmentation is an important step for image analysis and image interpretation, we focus on PCNN applications on image segmentation, establish a modified PCNN model, and propose a multi-threshold approach according to water valley area in histogram. Meanwhile, the adaptive determination method of PCNN parameters for image segmentation is presented.

2 PCNN Neuron Model

As showed in Fig.1, each PCNN neuron is divided into three compartments with characteristics of the receptive field, the modulation field, and the pulse generator.

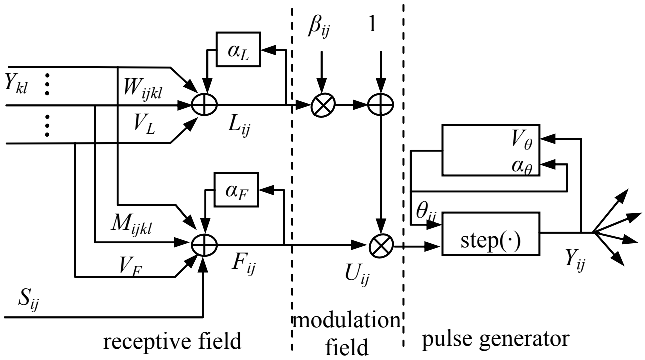


Fig. 1. Traditional PCNN neuron model

Each traditional PCNN neuron model has nine parameters to be determined, including three time decay constants (α_F, α_L, α_θ), three amplification factors (V_F, V_L, V_θ), linking coefficient β_{ij}, linking matrix *M* and *W*. The following five equations are satisfied.

$$F_{ij}(n) = \exp(-\alpha_F) \cdot F_{ij}(n-1) + S_{ij} + V_F \cdot \sum M_{ijkl} Y_{kl}(n-1) \quad . \tag{1}$$

$$L_{ij}(n) = \exp(-\alpha_L) \cdot L_{ij}(n-1) + V_L \sum W_{ijkl} Y_{kl}(n-1) \quad . \tag{2}$$

$$U_{ij}(n) = F_{ij}(n)(1+\beta_{ij}\cdot L_{ij}(n))\,. \tag{3}$$

$$\theta_{ij}(n) = \exp(-\alpha_\theta)\theta_{ij}(n-1) + V_\theta Y_{ij}(n-1) \quad . \tag{4}$$

$$Y_{ij}(n) = step(U_{ij}(n) - \theta_{ij}(n)) \quad . \tag{5}$$

Where *step*(•) is the unit step function. Moreover, to the whole neural network, the iteration times *N* should also be decided. The various parameters used in the PCNN model are of great significance when preparing the PCNN for a certain task.

The performance of segmentation results based on PCNN depends on the suitable PCNN parameters. It is necessary to determine the near optimal parameters of the network to achieve satisfactory segmentation results for different images. Up to now, the parameters are most adjusted manually and it is a difficult task to determine PCNN parameters automatically for different kinds of images.

3 Water Valley Area Based Adaptive Parameters Determination

3.1 Multi-threshold Approach Using Water Valley Area Method

In this paper, we propose the definition of 'water valley area' to determine multi-threshold in image segmentation. Assume *hist*(*f*(*x*,*y*)) is the histogram of image *f*(*x*,*y*); S_i (i=1,2,…,K) is the maximum points on *hist*(*f*(*x*,*y*)); Q_j (j=1,2,…,N) is the minimum points on *hist*(*f*(*x*,*y*)); P_m (m=1,2,…,M+1) is the peak points, which satisfied with $P_1 < P_2 < \ldots < P_{M+1}$; T_n (n=1,2,…,M) is the multi-thresholds, which satisfied with $T_1 < T_2 < \ldots < T_M$. P_m and T_n are unknown and waiting for solution. Obviously, $P \subseteq S$ and $T \subseteq Q$.

Defination (water valley, water valley area). Assume S_{i1} and S_{i2} is two maximum points of *hist*(*f*(*x*,*y*)), whose corresponding gray value is g_{Si1} and g_{Si2} respectively, and $g_{Si1} < g_{Si2}$. If there is no other maximum points in(g_{Si1}, g_{Si2}) or the value of existed maximum points is small than min{S_{i1}, S_{i2}}, we define a water valley between S_{i1} and S_{i2}. The bottom of water valley is the borderline of *hist*(*f*(*x*,*y*)), and the height of water valley is min{S_{i1}, S_{i2}}. Imagine we can use 'water' to abound the whole space, then the capacity can be defined as 'water valley area', *area*(g_{Si1}, g_{Si2}), the calculation formula is

$$area(g_{Si1}, g_{Si2}) = \frac{1}{2}\int_{g_{Si1}}^{g_{Si2}} \{[\min\{S_{i1}, S_{i2}\} - hist(x)] + |\min\{S_{i1}, S_{i2}\} - hist(x)|\} dx \tag{6}$$

Assume Q_j is the minimum point of (g_{Si1}, g_{Si2}), namely for $\forall g_x \in$ (g_{Si1}, g_{Si2}), $hist(g_x) \le Q_j$ is satisfied, we use *valley*(S_{i1}, Q_j, S_{i2}) to denote water valley.

The detailed process to get peak points and multi-thresholds is given below.

*Step1.*Draw image histogram *hist*(*f*(*x*,*y*))and smooth it to decrease noise influence if necessary.

Step2. Seek all extremum points in the histogram, including maximum points S_i (i=1,2,…,K) and minimum points Q_j (j=1,2,…, N). For the need of building water valley, the extremum points on two sides of *hist*(*f*(*x*,*y*)) must be maximum points, so $K = N + 1$.

Step3. From the left minimum point Q_1 and maximum points S_1、S_2 on its two sides($S_1<Q_1<S_2$), we built water valley, *valley*(S_l, Q_c, S_r)(l= 1, c=1, r=2), and calculate its area, A=*area*(g_{Sl}, g_{Sr}), by formula.(6).

Step4. Determine multi-thresholds and peak points, here, define Θ as a lower limitation ranging from 0.01 to 0.03. The smaller the value of Θ is, the more threshold points we will get.

(1) If $A \geqslant \Theta$, Q_c will be kept in threshold array T_n. Meanwhile, S_l will be kept in peak points array P_m. $S_l = S_r$, $Q_c = Q_r$, and $S_r = S_{r+1}$.

(2) If $A < \Theta$, the valley will be taken as invalid. At this situation, compare the value of S_l and S_r:

(i) if $S_l > S_r$, then S_l will be regarded as the new left maximum point, S_{r+1} is the new right maximum point. The smaller of Q_c and Q_r is minimum point in new water valley.

(ii) if $S_l \leq S_r$, then S_r、Q_r、S_{r+1} will be the left maximum point, minimum point and right maximum point of new water valley.

Step5. Calculate water valley area, A=*area*(g_{Sl}, g_{Sr}), by formula.(6) and iteratively execute step 4 until all minimum points have been processed.

At last, we can get the threshold array T_n (n=1,…M and $T_1<\ldots<T_M$) and the corresponding peak array P_m(m=1,…M+1 and $P_1<\ldots<P_{M+1}$). Hence, a valid water valley *valley*(P_m, T_m, P_{m+1}) includes two neighboring peaks $\{P_m, P_{m+1}\}$ and Fig.3 (c) shows water valleys and corresponding thresholds determined by this method.

3.2 Modified Pulse Coupled Neural Network

We have established a modified PCNN, which is implemented by applying iteratively the equations

$$L_{ij}[n] = \sum W_{ijkl} Y_{kl}[n-1] \tag{7}$$

$$U_{ij}[n] = S_{ij}(1 + \beta_{\mathrm{i,j}}[n] L_{ij}[n]) \tag{8}$$

$$Y_{ij}[n] = \begin{cases} 1, & U_{ij}[n] > T_{ij}[n] \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The indexes i and j refer to the pixel location in the image, indexes k and l refer to the dislocation in a symmetric neighborhood around a pixel, and n refers to the time (number of iteration). $L_{ij}[n]$ is linking from a neighborhood of the pixel at location (i,j), $U_{ij}[n]$ is internal activity at location (i,j) , which is dependent on the signal value S_{ij} at (i,j) and linking value. $\beta_{i,j}[n]$ is the PCNN linking parameter, and $Y_{ij}[n]$ is the output value of the PCNN element at (i,j). $T_{ij}[n]$ is a threshold value. We use a set of fixed threshold values, T_n(n=1,…M) determined by water valley area method mentioned above.

If $Y_{ij}[n]$ is 1 at location (i,j) at n=t, we say that the PCNN element at the location (i,j) fires at t. The firing due to the primary input S_{ij} is called the natural firing. The second type of firing, which occurs mainly due to the neighborhood firing at the previous iteration, we call the excitatory firing, or secondary firing.

Starting with the biggest threshold T_M, object whose mean gray value is larger than T_M will be picked out at the first iteration. We keep the threshold T_M fixed during the following iterations until no firing happens. At a certain threshold, the iteration times differ from image to image and a suitable amount of iterations in practice is 20-70. After the first iteration loop, both the natural firing pixels and excitatory firing pixels are collected, which is the first level PCNN segmented objects with the largest gray value. Then the second level objects can be got by the same algorithm using threshold T_{M-1}. Repeating this progress until all thresholds are processed, we will get M+1 levels of objects with different intensities at last.

In this PCNN algorithm, we are using the neighborhood with the radius r=1.5 (i.e., a usual 3×3 neighborhood, with the linking relative to the inverse of the squared distance from the midpixel and normalized to one.

Considering those pixels whose intensities are smaller than peak point P_m ought not to be captured at T_n even if they have the largest linking value 1, so in the iteration loop at T_n, the value of β_m is chosen to be

$$\beta_m = \frac{T_{\mathrm{n}}}{P_m} - 1 \tag{10}$$

Because P_1 may be 0, we choose the value of β_1 to be 0.1-0.3 at this situation.

4 Experiments

4.1 Compared with Current Typical Methods

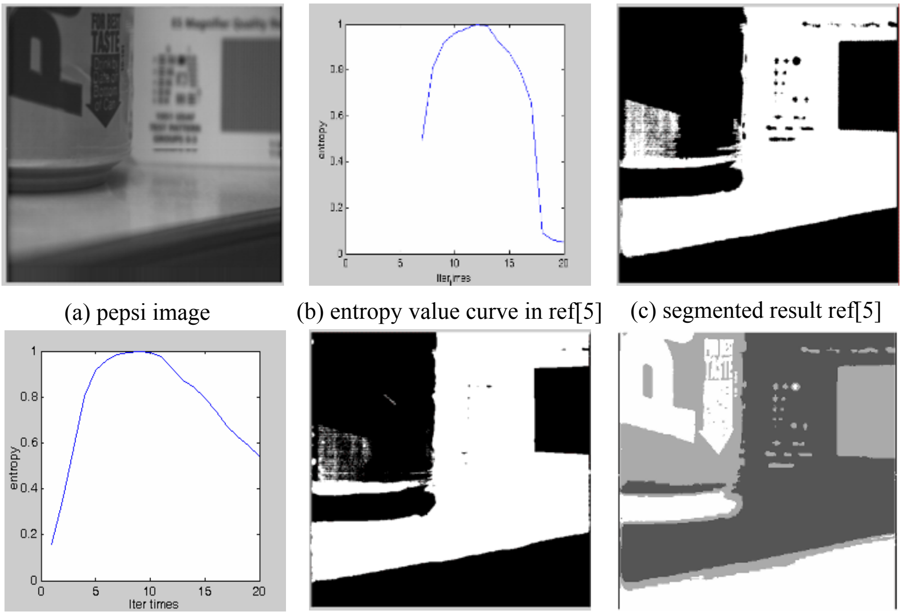
To evaluate the performance of the proposed method, we have compared with other typical PCNN parameters determination methods in image segmentation applications.

Gu X.D. *et al.* [5] have brought forward a new approach for image segmentation based on unit-linking PCNN. The main characteristic of the method is that linking L_{ij} is a binary function. The linking input is 1 if any neuron fire is in its nearest-neighbor 3×3 field, otherwise it is 0. As to optimum PCNN iteration times N, the maximal Shannon entropy method provided in reference [3] is used to determine it. We have segmented some images by this method and the results are not satisfying. Take the image in Fig.2 (a) as an example, Fig.2 (b) shows the binary segmented image's entropy value curve during iteration process, from which we can see the segmented image has the largest Shannon entropy 0.9989 when iteration times N is 12, the corresponding segmented result is showed as Fig.2 (c). Apparently, the segmented result has poor performance at that point.

Bi Y.W. and Qiu T.SH. [7] have brought forward a segmentation method based on a simplified PCNN with the parameters determined by images' spatial and grey characteristics automatically. Linking matrix M and W are determined by the pixel value distribution of central pixel neighbor r×r field and various from one another. Linking coefficient β_{ij} is defined as CV (Coefficient of Variation)

$$\beta_{ij} = CV_{ij} = \sqrt{V_{ij}} / M_{ij} \tag{11}$$

where V_{ij} and M_{ij} are the mean square deviation and mean gray value of pixel *(i,j)* neighbor field respectively. Threshold amplification factor is given a large value 50. Iteration times N is also determined by the maximal Shannon entropy method provided in reference [3]. Fig.2 (d) shows the binary segmented image's entropy value curve during iteration process, from which we can see the segmented image has the largest Shannon entropy 0.9995 when iteration times N is 9, the corresponding segmented result is showed as Fig.2 (e). Obviously, the segmented result is not satisfying too. Moreover, this method needs lots of calculations. Fig.2 (f) is the segmented image with our automatically parameters determination method based on the modified PCNN, from which we can see the performance of our method outperforms current methods greatly.



(a) pepsi image (b) entropy value curve in ref[5] (c) segmented result ref[5]

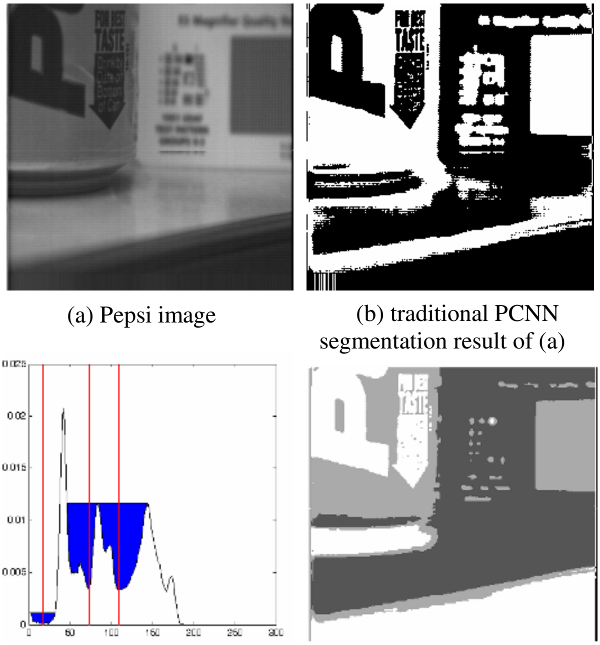
(d) entropy value curve in ref[7] (e) segmented result ref[7] (f) segmented result by the proposed method

Fig. 2. Compared with other typical methods

4.2 Compared with Traditional PCNN Performance

In order to comparing with the performance with traditional PCNN (showed as Fig.1), the experiments that PCNN used in image fusion applications are also carried out.

As Fig.3 shown, Fig.3 (a) is source images. Fig.3 (b) is the segmented results by traditional PCNN. Fig.3 (c) shows the corresponding water valleys and multi-thresholds determined by water valley area. The segmented result by modified PCNN, which parameters are determined by water valley area, is showed as Fig.3 (d). Table.1 shows the parameters of traditional PCNN which were tuned by trial in order to get perfect segmentation performance.



(a) Pepsi image

(b) traditional PCNN segmentation result of (a)

(c) water valleys and multi-hresholds of (a)

(d) modified PCNN segmentation result of (a)

Fig. 3. The "Pepsi" source images (256 level, size of 512×512) and segmented results

Table 1. Values of parameters in image segmentation by traditional PCNN

parameters	β	α_F	α_L	α_θ	V_F	V_L	V_θ	r	N
Fig.3(b)	0.3	1	4	2	10	10	100	1	2

From Fig.3, we can see the image segmented by the proposed method provides more details and useful information. The idea of multi-threshold makes segmented image more levels than traditional PCNN. We must mention that comparing with the parameters of traditional PCNN, which were tuned by trial, the parameters in our method can be determined automatically, this has great importance in expanding the application range of PCNN.

5 Conclusion

In order to determine PCNN parameters adaptively, this paper brings forward an adaptive segmentation algorithm based on a modified PCNN with the multi-thresholds determined by water valley area method. The main contributions include establishing a modified PCNN, proposing adaptive PCNN parameters determination algorithm based on water valley area, and implementing the described methods on PCNN applications. Experimental results show its good performance and robustness.

The research fruits have great importance both on the theory research and practical application of PCNN.

References

1. Eckhorn, R., ReitBoeck, H.J., et al.: Feature linking via synchronization among distributed assemblies: simulation of results form cat visual cortex. Neural Computation, Vol. 2. (1990) 293−307
2. Kuntimad, G., Ranganath, H.S.: Perfect image segmentation using pulse coupled neural networks. IEEE Trans. Neural Networks, Vol.10. (1999) 591−598
3. Ma, Y.D., Dai, R.L., Li, L.: Automated image segmentation using pulse coupled neural networks and image's entropy. Journal of China Institute of Communications, Vol.23. (2002) 46−51
4. Liu, Q., Ma, Y.D., Qian, ZH.B.: Automated image segmentation using improved PCNN model based on cross-entropy. Journal of Image and Graphics, Vol.10. (2005) 579−584
5. Gu, X.D., Guo, Sh.D, Yu, D.H.: A new approach for automated image segmentation based on unit-linking PCNN. Proceedings of the first International Conference on Machine learning and Cybernetics, Beijing, China, (2002) 175−178
6. Karvonen, J.A.: Baltic sea ice SAR segmentation and classification using modified pulse-coupled neural networks. IEEE Trans. Geoscience and Remote Sensing, Vol.42. (2004) 1566−1574
7. Bi, Y.W., Qiu, T.SH.: An adaptive image segmentation method based on a simplified PCNN. ACTA ELECTRONICA SINICA, Vol.33. (2005) 647−650
8. Aboul, E.H., Jafar, M.A.: Digital Mammogram Segmentation Algorithm Using Pulse Coupled Neural Networks. Proceedings of the 3th International Conference on Image and Graphics, Hong Kong, China, (2004) 92−95
9. Gu, X.D., Yu, D.H.: Image shadow removal using pulse coupled neural network. IEEE Trans. Neural Networks, Vol.16. (2005) 692−695
10. Ma, Y.D., Shi, F, Li L.: Gaussian noise filter based on PCNN. In: IEEE Int. Conf. Neural Networks & Signal Processing, Nanjing, China, (2003)149–151
11. Johnson, J.L., Taylor, J.R., Anderson, M.: Pulse coupled neural network shadow compensation. SPIE Conference on Applications and Science of Computational Intelligence, Orlando, Florida, (1999) 452–456
12. Ekblad, U., Kinser, J.M., Atmer, J., Zetterlund, N.: The intersecting cortical model in image processing. Nuclear Instruments and Methods in Physics Research A 525, (2004) 392–396

The Forgetting Gradient Algorithm for Parameter and Intersample Estimation of Dual-Rate Systems

Yang Hui-zhong*, Tian Jun, and Ding Feng

Research Center of Control Science and Engineering,
Southern Yangtze University, 214122, Wuxi, P.R. China
sytutianjun@yahoo.com.cn

Abstract. Multirate systems are abundant in process industry, many soft-sensor design problems are related to modeling, parameter identification, or state estimation involving multirate systems. In this paper, a polynomial transformation technique has been used to derive a dual-rate model with a finite number of parameters; based on this model, the dual-rate forgetting gradient algorithm has been used to estimate the model parameters and intersample outputs based on the dual-rate input-output data directly. Furthermore, convergence properties of the algorithms in the stochastic framework are studied and show that 1) the parameter estimation error consistently converges to zero under the persistent excitation condition; 2) the intersample output estimation error is uniformly bounded. Finally, a simulation example show excellent effectiveness in parameter and output estimation.

1 Introduction

This paper deals with a class of multirate systems-the dual-rate systems as shown in Fig. 1, where P_c is assumed to be a continuous-time process with an additive disturbance $v(t)$; the input to P_c is produced by a zero-order hold H_T with period T, processing a discrete-time signal $u(kT)$; $y_0(t)$ is the noise-free output or true output of P_c but unmeasurable; the output $y(t)$ of P_c is sampled by a sampler S_{qT} with period qT. The available on-line input-output measurement data are:

- $\{u(k): \quad k = 0, 1, 2, \cdots\}$ at the fast rate, and
- $\{y(kq): \quad k = 0, 1, 2, \cdots\}$ at the slow rate.

T is the basic sampling period and q is any finite positive integer. For notational simplicity, $T = 1$ in the following discussion.

Such multirate systems exist widely in process industries, many soft-sensor design problems are related to modeling, parameter identification, or state estimation involving multirate systems. For example, in polymer reactors[1,2], the

* This research was supported by the the National Natural Science Foundation of China (No. 60574051).

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 721–728, 2006.

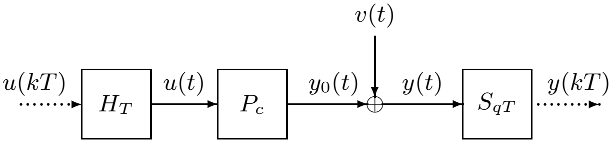


Fig. 1. The dual-rate systems

composition, density or molecular weight distribution measurements are typically obtained after several minutes of analysis, whereas the manipulated variables can be adjusted at relatively fast rate. Model identification and intersample output estimation in such multirate framework are important in that using these can monitor the output variables(which are sampled infrequently due to hard limits on sensoring devices[3-5]) between samples, performing inferential control[6] and self-turning control[7].

In the process identification literature, Lu and Fisher used projection and least-squares based algorithms for estimating intersample outputs [8,9]; but their algorithms handle only noise-free dual-rate systems. Ding proposes dual-rate least-squares(DR-LS) algorithms for various system model based on the polynomial transformation technique in stochastic framework[10-12]. Although the DR-LS may be used to identify a dual-rate model, but this model has more parameters than the original system, especially for large q; hence the corresponding algorithm requires a large amount of computation. The objective of this paper is to provide a DR-FG algorithm to estimate the parameters of the dual-rate models and the intersample outputs based on dual-rate data directly.

2 Modeling of Dual-Rate Systems

Fig. 1 is a simple dual-rate system, where H_T is a zero-order holder with period T, S_{qT} a sampler with period qT ($q > 2$ being an integer). For convenience, writing $u(k) := u(kT), y(kq) := y(kqT)$. Thus, the intersample outputs (also called missing outputs), $y(kqT+iT) =: y(kq+i), i = 1, 2, \cdots, q-1$ are unavailable due to hardware limitation. Therefore, the objectives of modeling and identification of multirate systems are two parts: 1) to establish the mapping relationship between available input and output data, 2) to estimate the intersample (missing) outputs by using the obtained model.

The open-loop transfer function from $u(k)$ to $y(k)$ takes the following real-rational form:

$$P_1(z) = \frac{b(z)}{a(z)}, \quad \text{or} \quad y(k) = \frac{b(z)}{a(z)}\,u(k) \tag{1}$$

with

$$a(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n},$$
$$b(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n}.$$

But the model in (1) is not appropriate for dual-rate system identification. Therefore, $P_1(z)$ needs to be transformed into a form with which directly uses the dual-rate data. A polynomial transformation technique[11,12] can be adopted to do this. The details are as follows.

Let the roots of $a(z)$ be z_i, then

$$a(z) = \prod_{i=1}^{n}(1 - z_i z^{-1}).$$

Define

$$\phi_q(z) := \prod_{i=1}^{n}(1+z_i z^{-1}+z_i^2 z^{-2}+\cdots+z_i^{q-1} z^{-q+1}) = \prod_{i=1}^{n}\frac{1 - z_i^q z^{-q}}{1 - z_i z^{-1}}.$$

Multiplying the numerator and denominator of $P_1(z)$ by $\phi_q(z)$, transforming the denominator of $P_1(z)$ into the desired form:

$$P_2(z) = \frac{b(z)\phi_q(z)}{a(z)\phi_q(z)} =: \frac{\beta(z)}{\alpha(z)} \tag{2}$$

with

$$\alpha(z) = a(z)\phi_q(z) = 1 + \alpha_1 z^{-q} + \cdots + \alpha_n z^{-qn} \tag{3}$$

$$\beta(z) = b(z)\phi_q(z) = \beta_0 + \beta_1 z^{-1} + \cdots + \beta_{qn} z^{-qn} \tag{4}$$

In this way obtaining the desired dual-rate transfer function model in (2). Of course, the two models in (1) and (2) are equivalent: the one in (1) with $a(z)$ and $b(z)$ is simpler, and the one in (2) with $\alpha(z)$ and $\beta(z)$ is more complicated due to the common factor $\phi_q(z)$. However, the advantage with the model in (2) is that the denominator is a polynomial of z^{-q}; arising from here is a recursive equation using only slowly sampled outputs.

3 The Parameter Estimation and Output Estimation Algorithms

In this section, the parameter and intersample output estimation problem using the model in (2) in the stochastic framework is studied. Based on the model in (2) and introducing a noise term $v(k)$, giving

$$\alpha(z)y(k) = \beta(z)u(k) + v(k)$$

where $v(k)$ is assumed to be a zero-mean random signal. Substituting the polynomials $\alpha(z)$ in z^{-q} in (3) and $\beta(z)$ in z^{-1} in (4) leads to the following regression equation,

$$y(k) = \varphi^T(k)\theta + v(k) \tag{5}$$

where the superscript T denotes the matrix transpose, and the parameter vector θ and information vector $\varphi(k)$ are defined by

$$\theta = [\alpha_1, \alpha_2, \cdots, \alpha_n, \beta_0, \beta_1, \cdots, \beta_{qn}]^T \in R^N, N = qn + n + 1$$

$$\varphi(k) = [-y(k-q), \cdots, y(k-qn), u(k), u(k-1), \cdots, u(k-qn)]^T$$

Here θ contains all parameters in the model in (5) to be estimated, and $\varphi(k)$ uses only available dual-rate data – if k is an integer multiple of q, then $\varphi(k)$ contains only the past measurement outputs (slow rate) and past and (possibly) current inputs (fast rate). Replacing k in (7)with kq gives

$$y(kq) = \varphi^T(kq)\theta + v(kq) \tag{6}$$

Let $\hat{\theta}(kq)$ be the estimate of θ at time kq. The following stochastic gradient algorithm for estimating the parameter vector θ of the dual-rate system in (6)(DR-SG for short).

$$\hat{\theta}(kq) = \hat{\theta}(kq-q) + \frac{\varphi(kq)}{r(kq)}[y(kq) - \varphi^T(kq)\hat{\theta}(kq-q)] \tag{7}$$

$$\hat{\theta}(kq+i) = \hat{\theta}(kq), i = 1, 2, \cdots, q-1$$

$$r(kq) = r(kq-q) + \|\varphi(kq)\|^2, \quad r(0) = 1 \tag{8}$$

where $\hat{\theta}(kq)$ is the estimation of θ in kq, and $\theta(\hat{0})$ some small real vector. The norm of matrix x was defined as $\|X\|^2 = tr(XX^T)$. Notice that the paratemer estimate $\hat{\theta}$ is updated every q samples, namely, at the slow rate; between the slow samples, keeping $\hat{\theta}$ unchanged. Thus, when having q new input samples and one new output sample, $\hat{\theta}$ is updated once.

The intersample outputs can be estimated as follows:

$$\hat{y}(kq+i) = \begin{cases} y(kq) \ , & i = 0 \\ \hat{\varphi}^T(kq+i)\hat{\theta}(kq), & i = 1, 2, \ldots, q-1 \end{cases}$$

where

$$\hat{\varphi}(kq+i) = [-\hat{y}(kq-q+i), -\hat{y}(kq-2q+i) \cdots - \hat{y}(kq-qn+i)$$

$$u(kq+i), u(kq+i-1) \cdots u(kq+i-qn)]^T$$

Another important work in this paper is the convergency property of the parameter estimate, as well as how to bound the intersample output estimation error, if the model is used to estimate missing output samples.

4 Convergence of the Parameter and Output Estimation

Theorem 1. For the dual-rate system in (6) and the DR-SG algorithm, assume that $\{v(k), \mathcal{F}_k\}$ is a martingale difference sequence defined on a probability space $\{\Omega, \mathcal{F}, P\}$, where $\{\mathcal{F}_k\}$ is the σ algebra sequence generated by $\{v(k)\}$, i.e., $\mathcal{F}_k = \sigma(v(k), v(k-1), v(k-2), \cdots)$, and that the noise sequence $\{v(k)\}$ satisfies the following conditions:

$$1) E[v(k)|\mathcal{F}_{k-1}] = 0, a.s$$

$$2) E[v^2(k)|\mathcal{F}_{k-1}] = \sigma_v^2(k) \leq \bar{\sigma}_v^2 < \infty, a.s$$

$$3) \limsup_{k\to\infty} \frac{1}{k} \sum_{i=1}^{k} v^2(i) \leq \bar{\sigma}_v^2 < \infty, a.s$$

where "a.s" is "almost surely". If the SPE condition in (*) holds. Then the parameter estimation $\hat{\theta}(kq)$ given by the DR-SG algorithm converges to true parameter θ. That is

$$\lim_{k\to\infty} \hat{\theta}(kq) = \theta$$

In order to improve the convergence rate and, a forgetting factor λ is introduced getting the following DR-FG algorithm:

$$\hat{\theta}(kq) = \hat{\theta}(kq-q) + \frac{\varphi(kq)}{r(kq)}[y(kq) - \varphi^T(kq)\hat{\theta}(kq-q)]$$

$$\hat{\theta}(kq+i) = \hat{\theta}(kq), i = 1, 2, \cdots, q-1$$

$$r(kq) = \lambda r(kq-q) + \|\varphi(kq)\|^2, \quad r(0) = 1, 0 \leq \lambda \leq 1,$$

When $\lambda = 1$, the DR-FG algorithm reduces to the DR-SG algorithm; when $\lambda = 0$, the DR-FG algorithm is the dual-rate projection algorithm.

The following theorem gives convergence of the intersample output estimates.

Theorem 2. For the dual-rate system in (6) and the DR-FG algorithm , assume that $\alpha(z)$ is strictly stable, i.e., all zeros of $\alpha(z)$ are strictly inside the unit circle. Then the bounded input assumption implies that the output estimation error $\eta(kq+i) = \hat{y}(kq+i) - y(kq+i)$ is bounded, i.e,

$$\lim_{k\to\infty} \frac{1}{k} \sum_{i=k_0}^{k} E[\eta^2(i)|\mathcal{F}_{k-1}] \leq \sigma_v^2, \quad for\ any\ k_0 < \infty.$$

Proof. Because $\alpha(z)$ is strictly stable, there exists an integer k_0 such that for any $k \geq k_0$, $\hat{\alpha}(kq, z)$ is also stable, and $\hat{\varphi}(k)$ is bounded, i.e.,

$$\|\hat{\varphi}(k)\|^2 \leq \delta_{\hat{\varphi}} < \infty, \ , \quad \text{for any} \quad k \geq k_0.$$

From the definitions of $\eta(kq+i)$ and $\hat{y}(kq+i)$, having

$$\alpha(z)\eta(kq+i) = \alpha(z)\hat{y}(kq+i) - \alpha(z)y(kq+i) = \hat{\varphi}^T(kq+i)\tilde{\theta}(kq) - v(kq+i).$$

Since $\eta(kq) = 0$, applying Lemma B.3.3 in (13), there existing constants $c_5 < \infty$ and $c_6 < \infty$ such that for all $k \geq k_0$,

$$\sum_{i=k_0}^{k} \eta^2(i) \leq c_5 \sum_{i=k_0}^{k} [\hat{\varphi}^T(i)\tilde{\theta}(i) - v(i)]^2 + c_6 \leq 2c_5 \sum_{i=k_0}^{k} [\delta_{\hat{\varphi}} \|\tilde{\theta}(i)\|^2 + v^2(i)] + c_6$$

Taking expectation, dividing k and using 3) yield

$$\frac{1}{k} \sum_{i=k_0}^{k} E[\eta^2(i)|\mathcal{F}_{k-1}] \approx \sigma_v^2$$

5 Examples

Assume that the discrete system model takes the following form

$$P_1(z) = \frac{b(z)}{a(z)} = \frac{0.412 + 0.309z^{-1}}{1 - 1.60z^{-1} + 0.80z^{-2}},$$

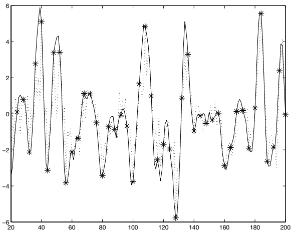
Taking $q = 4$, i.e., $\{u(k), y(4k)\}$ are available data. The corresponding dual-rate model with additive white noise can be expressed as Here $\{u(k)\}$ is taken as a persistent excitation sequence with zero mean and unit variance, and $\{v(k)\}$ as a white noise sequence with zero mean and variance σ_v^2. Applying the DR-FG algorithm to estimate the parameters (α_i, β_i) of this system. The parameter estimates are shown in Table 1, where δ is the relative parameter estimation error measured in the Euclidean norm:$\delta = \|\hat{\theta}(k) - \theta\|/\|\theta\|$, $\delta_a = \|\hat{\theta}(kq) - \theta\|$ is the absolute parameter estimation error. From Table 1, it is clear that δ, δ_a are becoming smaller as k increases.

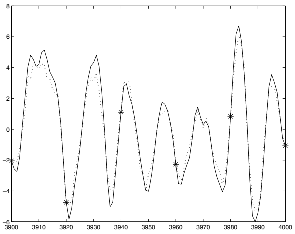
Fig. 2 illustrates a simulation for $q = 4$, where the whole transient is shown and the output estimation error approaches state before $200T$, i.e., before 50

Table 1. The DR-FG estimate of parameter($\lambda = 0.5, \sigma_v^2 = 1.00$)

k	100	500	1000	2200	θ
α_1	0.44378	0.22655	0.34817	0.36062	0.3584
α_2	0.25625	0.41595	0.40505	0.42836	0.4096
β_1	0.26824	0.41349	0.43763	0.41775	0.412
β_2	0.617	0.95285	0.93667	0.98897	0.9682
β_3	0.6001	1.1282	1.2154	1.1805	1.2195
β_4	0.55817	1.1881	1.1659	1.1512	1.1767
β_5	0.28193	0.93775	1.0333	1.0178	1.0547
β_6	0.11996	0.72378	0.82649	0.86716	0.85696
β_7	0.30649	0.43419	0.4812	0.45952	0.52736
β_8	0.31898	0.045062	0.102407	0.16193	0.15821
δ	0.11556	0.012301	0.00199	0.0012626	
δ_a	0.73837	0.079035	0.012789	0.0081123	

measured values of output are sampled. Note that this involves only 50 measured values of $y(kqT)$. Fig. 3 illustrates another simulation run for $q = 20$, where the output estimation error has essentially been eliminated after time $3900T$. For all these simulation runs $\hat{\theta}(0) = 0$ and the parameter estimates converge to the true parameters of the equivalent model (2). Fig.4 shows the δ_{ns} with different λ respectively.





Solid line:$y(k)$, dotted:$\hat{y}(k)$, * sample instant

Fig. 2. Output estimation (q=4)

Fig. 3. Output estimation(q=20)

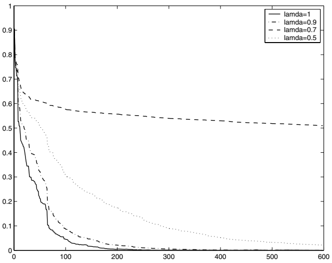


Fig. 4. δ_{ns} with different λ

6 Conclusions

A recursive DR-FG algorithm of identifying dual-rate systems is presented when the output is sampled q times slower than the input; the algorithm uses only dual-rate measurement data. Convergence performance of the proposed estimation algorithms is analyzed in detail in the stochastic framework. Based on the estimated models, intersample output estimation is also studied. It was shown that, the intersample output estimation error is bounded and converges to nearly zero. This formulation and proof provides a basis for multirate sampling applications in inferential, time-delay, and adaptive control. Although the analysis in the paper is done for dual-rate equation-error models with an additive white noise, the methods developed can be easily extended to dual-rate stochastic systems with colored noise.

References

1. Ohshima, M., Hashimoto, I., Takeda, M., Yoneyama, T.: Multirate multivariable model predictive control and its application to a semi-commercial polymerization reactor. Proceedings of the 1992 American Control Conference. Chicago. USA. (1992) 1576-1581
2. Gudi, R.D., Shah, S.L., Gray, M.R.: Multirate state and parameter estimation in an antibiotic fermentation with delayed measurements. Biotechnology and Bioengineering 44 (1994) 1271-1278
3. Li, D.G., Shah, S.L., Chen, T. Qi, K.Z.: Application of dual-rate modeling to CCR octane quality inferential control. IEEE Transactions on Control Systems Technology. 11 (2003) 43-51
4. Gudi, R.D., Shah, S.L., Gray, M.R.: Adaptive multirate state and parameter estimation strategies with application to a bioreactor. AIChE J. 41 (1995) 2451-2464
5. Huang, Z.W., Xu, Y.M., Fang, C.Z., Tang, J.: Improvements in dynamic compartmental modelling of distillation columns. Journal of Process Control. 3 (1993) 139-145
6. Li, D., Shah, S.L., Chen, T.: Analysis of dual-rate inferential control systems. Automatica. 38 (2002) 1053-1059
7. Ding, F., Chen, T.: Adaptive control of dual-rate systems based on least squares methods. In: Proceeding s of 2004 American Control Conference (ACC). Boston. Massachusetts. USA. (2004) 3508-3513
8. Lu, W.P., Fisher, D.G.: Least-squares output estimation with multirate sampling. IEEE Transactions on Automatic Control. 34 (1989) 669-672
9. Lu, W.P., Fisher, D.G.: Output estimation with multi-rate sampling.International Journal of Control. 48 (1998) 149-160
10. Ding, F., Chen,T.: Parameter estimation of dual-rate stochastic systems by using an output error method. IEEE Transactions on Automatic Control. 50 (2005) 1436-1441
11. Ding, F., Chen,T.: Identification of dual-rate systems based on finite impulse response models. International Journal of Adaptive Control and Signal Processing. 18 (2004) 589-598
12. Ding, F., Chen,T.: Modeling and identification for multirate systems. Acta Automatica Sinica, 31 (2005) 105-122
13. Ding, F., Chen,T.: Parameter identification and intersample output estimation of a class of dual-rate systems. The 42nd Conference on Decision and Control. December 9-12. 2003. Hyatt Regency Maui,Hawaii,USA.
14. Ding, F., Yang, J.: The convergence analysis of stochastic gradient algorithm. Journal of Tsinghua University. 39 (1999) 83-86
15. Xie, X., Ding, F.: Adaptive control system. BeiJing: Tingshua University press, 2002

Intelligent System for Feature Extraction of Oil Slick in SAR Images: Speckle Filter Analysis

Danilo L. de Souza[1], Adrião D.D. Neto[1], and Wilson da Mata[2]

[1] Federal University of Rio Grande do Norte, Department of Computing Engineering and Automation, Natal, RN, 59072-970, Brazil
danilo@dca.ufrn.br, adriao@dca.ufrn.br
[2] Federal University of Rio Grande do Norte, Department of Electrical Engineering, Natal, RN, 59072-970, Brazil
wilson@ct.ufrn.br

Abstract. The development of automatic techniques for oil slick identification on the sea surface, captured through remote sensing images, cause a positive impact to a complete monitoring of the oceans and seas. C-band SAR (ERS-1, ERS-2, Radarsat and Envisat projects) is well adapted to detect ocean pollution because the backscatter is reduced by oil slick. This work propose a system for segmentation and feature extraction of oil slicks candidates based on techniques of digital image processing (filters, gradients, mathematical morphology) and artificial neural network (ANN). Different algorithms of speckle filtering are tested and a comparison for the considered system is presented. The process is thought to possess a level of automatization that minimizes the intervention of a human operator, being possible the processing of larger amount data. The focus of the work is to present a study detailed for feature extraction block proposed (architecture used and computational tools).

1 Introduction

Oil spills on the sea surface damage the marine ecosystem, specially when they occur next to coast. With regular passing over the seas and oceans, the imagery satellites furnish data which may be used on the extraction of statistical information about spots of many specific regions on the Earth. The actual techniques of oil slick identification use SAR (Synthetic Aperture Radar) images.

The presence of oil film on the sea surface damps the small waves due to the increased viscosity of the top layer and drastically reduces the measured backscattering energy, resulting in darker areas in SAR imagery. However, careful interpretation is required because dark areas in SAR images might also be caused by locally low winds, by natural sea slicks or internal waves , all of them called "look-alikes" [1].

Some approaches exist for oil slick analysis from satellite images [1][2][3], however automatic analysis of SAR images is not applied routinely yet. The main research is on classifier algorithm, but a consistent block for feature extraction is not encoutered in literature. The systems that use spot's feature for the classification, basically, follow the same structure, however none of them describes the algorithms used in feature extraction block, as well as the robustness of the process.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 729–736, 2006.

The focus of the work is to present a detailed study for feature extraction block (architecture used and computational tools). The section 2 presents the process of oil slick detection. In this section, both, the tool of digital image processing and the features of interest are presented in feature extractor block. In section 3 an experimental result is showed whith a SAR image. Section 4 summarizes the conclusions and the future work.

2 Feature Extraction

The process for oil slick detection is showed in figure 1, the arrows indicate the information flux. A human operator, who selects the interest area (dark spot) to be submitted to the system, carries through the analysis of the image. The next stage computes the features for the classifier, that estimate the probability of the spot to be a slick.

Some of the describers which can be extracted from sensors SAR images, in the characterization of spots on the sea surface for varied satellites, are detailed by Solberg et al. [4][5] and Del Frate et al. [1][6].

The features in this work are of three different types. Some of them contain information of the backscattering intensity gradient along the border of the analysed dark spot: Maximum Gradient (Gmax), Mean Gradient (Gme), Gradient Standard Deviation (GSd), all in dB; others focus on the backscattering in dark spots and/or in the background: Object Standard Deviation (OSd), Background Standard Deviation (BSd), Maximum Contrast (ConMax), Mean Contrast (ConMe), also in dB; a third category takes into account the geometry and the shape of the dark spots: Area (A) in km^2, Perimeter (P) in km, shape Complexity (C), Spreading (S) [1][7].

2.1 Speckle Filtering

A good filtering of the noise speckle must preserve the backscatter average values, besides keeping well definite edges between adjacent fields, and still preserving the space variability (textural information) related to the scene.

The median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. The median is calculated by replacing the pixel being considered with the middle pixel value. This filter possesses the trend of better preserving the texture information, however does not preserve the signature of prompt targets [8].

- Lee. It adopts a multiplicative model for the noise and obeys to the criterion of “local linear minimum mean square error”. It minimizes the quadratic average error through the Wiener filter. The Lee filter is a adaptive and general filter [9].
- Frost. It is a linear convolutional filter, derivative of the minimization of mean square error on the multiplicative model of the noise. It is an adaptive filter that preserves the structure of edges [9].
- Kuan. It adopts the multiplicative model. The procedure is similar to that one of Lee. It is also an adaptive and general filter [9].

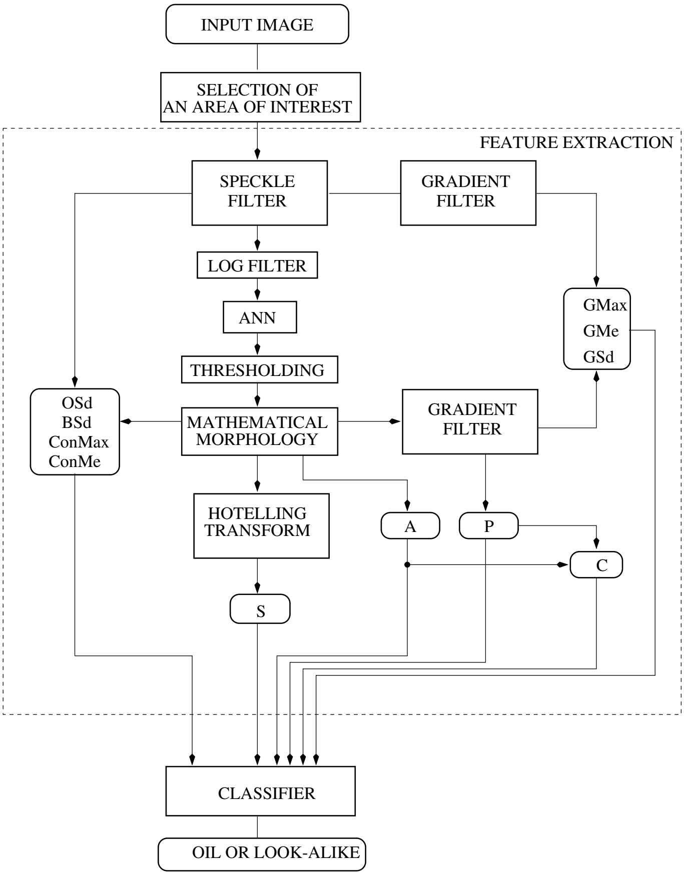


Fig. 1. System for oil slick detection. Feature extractor block detailed.

Lopes et al. [10] propose to divide an image into areas of three classes. The first class correponds to the homogeneous areas in which the speckles may be eliminated simply applying a Low Pass filter. The second class corresponds to the heterogeneous areas in which speckles are to be reduced while preserving texture. And the third class, are areas containing isolated point targets, which, in this case, the filter should preserve the observed value.

Based on the above considerations, Lopes et al. [10] modified the Lee, and the Frost filters, labeled here as Enhanced Lee filter and Enhanced Frost filter, respectively. The input data for this block is a fragment of SAR image selected.

2.2 Gradient Filter

The presence of a tone difference between background and object turn possible to gradient operator:

- To measure this difference, variation of tone. This information is important because in case of look-alikes the values of backscattering either in the object or in the surroundings are more dispersed. Also, oil spills show mean value of the gradient along the border higher than look-alikes.
- To label the pixels which belong the edge of the object in study.

The input data for this two blocks are the output images of speckle filter and mathematical morphology blocks. The output data are the features: Maximum Gradient (Gmax), Mean Gradient (Gme), Gradient Standard Deviation (GSd) and Perimeter (P).

2.3 Log Filter

The realce of the image by logarithmic function has as property a mapping close to the linear for small values of input. This is interesting therefore our object of study is composed of dark tonalities. Therefore for high values of input the logarithmic function tends to map the output for high values, however also close.

This condition preserves the grey levels whose values are similars to the spot's value, thus minimizing the creation of spurious regions in the processed image. Moreover it makes possible the reduction of the time in artificial neural network block, speeding the process.

2.4 Artificial Neural Network – ANN

In the unsupervised neural network the learning process is given to carry through a measure, independent of the task, the quality of the representation that the net must learn, and the free parameters of the net are optimized in relation to this measure. A time that the net it's self-adjusted to the statistical regularity of the input data, it develops the ability to form internal representations to codify the characteristics of the input and, in this way, to create new classes automatically [11].

To carry through the not supervised learning, we use the competitive learning rule, implementing a ANN with two layers, one of input and one competitive. The input layer receives the data available. The competitive layer is composed for neurons which the units compete for the exclusive right to respond to a particular input pattern. The neural network, operates a strategy of the type "the winner takes all" [11].

In a "winner takes all" strategy, the output unit k receiving the largest induced local field v_k, for a specified input pattern $\mathbf{x}$, is assigned a full value 1, whereas all other units are suppressed to a 0 value.

$$y_k = \begin{cases} 1 \text{ if } v_k > v_j & \text{for all} \quad j, j \neq k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

In accordance with the standard rule of the competitive learning, the variation Δw_{kj} applied to the synaptic weight w_{kj} is defined by the equation (2).

$$\Delta w_{kj} = \begin{cases} \eta\,(x_j - w_{kj}) & \text{if the } k \text{ neuron wins} \\ 0 & \text{if the } k \text{ neuron loses} \end{cases} \tag{2}$$

where, η is the learning rate . This rule has the global effect that it moves the synaptic weight vector $\mathbf{w_k}$ from winning neuron k to the input pattern $\mathbf{x}$, into the region where the actual stimuli lie.

The input parameters of this block are the output image from log filter block and the number of neurons or classes in the competitive layer.

2.5 Thresholding

The thresholding is extremely quick, and generally carried out in one step only, respecting the following rule, in the case of a binary partition [8]:

$$\begin{cases} \text{if} \;\; f\left(x,y\right) > t \longrightarrow f\left(x,y\right) \in \text{background} \\ \text{else} \qquad\qquad\;\; \longrightarrow f\left(x,y\right) \in \text{spot} \end{cases} \tag{3}$$

where t is a key parameter, because it determines the excellent partition of the classes. The choice of the parameter t in equation (3) considers the only known priori information: the spot is composed for dark tones. Then, the value of t can be gotten automatically through the comment of the lesser value of synaptic weight of the neurons of the unsupervised net , in the previous process step. The input parameter for this block is the output image of the ANN block.

2.6 Mathematical Morphology

The Mathematical Morphology stage completes the process of spot segmentation. The base of the mathematical morphology is the theory of sets, which represents the forms of objects in an image. The objective consists of: to extract from an unknown set (image) morphologic information from the use of another completely definite set, called structuring element.

The choice of the structuring element can be carried through some forms. In many cases, literature suggests forms known for specific applications. The use of the morphologic opening operator has the objective to eliminate “islands”. The morphologic opening is composed for the erosion operation followed by the dilation operation, always with the same structuring element.

The input parameter for this block is the binary image from thresholding block. The output data of this block are the characteristics: Object Standard Deviation (OSd), Background Standard Deviation (BSd), Maximum Contrast (ConMax), Mean Contrast (ConMe), Area (A) and jointly with the calculated Perimeter previously we get the Complexity (C).

2.7 Hotelling Transform

The objective of the transform is the representation of the original data set, in a set of plans of main components, where the information is organized by the relevance degree that it brings of the original data set [8].

The visualized object is treated as a bidimensional set. Each pixel in the object is a bidimensional vector $\mathbf{p}$, where p_1 and p_2 are the coordinates corresponding to axes of image. These vectors are used to compute an average vector, $\mathbf{m_p}$, and a covariance matrix, $\mathbf{C_p}$, of the set.

A transformation matrix $\mathbf{A}$ whose rows are formed from the eigenvectors of $\mathbf{C_p}$, arranged that its first row is the eigenvector corresponding to the eigenvalue greater, and its last row is the eigenvector corresponding to the eigenvalue minor. The matrix $\mathbf{A}$ mapping vectors p in vectors q, as follows:

$$\mathbf{q} = \mathbf{A}\left(\mathbf{p} - \mathbf{m_p}\right) \tag{4}$$

The equation (4) is called Hotelling Transform. A covariance matrix $\mathbf{C_q}$ can be extracted of the vectors $\mathbf{q}$.

$$\mathbf{C_q} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \tag{5}$$

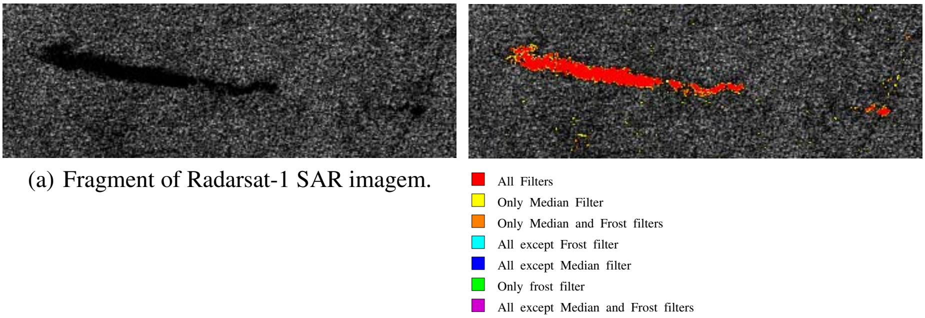
The elements out of the $\mathbf{C_q}$ diagonal possess value 0, which show that the components of $\mathbf{q}$ vectors are not correlated [8]. The eigenvalues of $\mathbf{C_q}$ are used to get the last spot's feature, the Spreading (S). This measure is a rate between the variance from main axis and the secondary axis of the spot.

3 Experimental Results

We propose to analyse two images from different imagery satelites. The first is a fragment of Radarsat-1 SAR image, from May, 21^{th}, 1999, 08:10:33 UTM (orbit 0018490), showed in figure 2(a). The second is a fragment of ERS-2 SAR image, February, 04^{th}, 2002, 11:03:41 GM (orbit 35519 frame 2885), showed in figure 3(a).

It was applied the algorithms presented in section 2. The mask's size of each filter tested was defined as 3×3. The number of neurons on the competitive layer was chosen for a clean sea area, an oil area and an intermediate one between these two first classes, the emulsion, with 3 specialysed neurons. For gradient block, the Sobel operator has been implemented and, initially, we opt by structuring element known as "cross" for mathematical morphology block.

In the presented results for Radarsat-1 image, the adaptive filters (Frost, Lee, Kuan, Enhanced Frost and Enhanced Lee) had behaved in very similar way, however they had



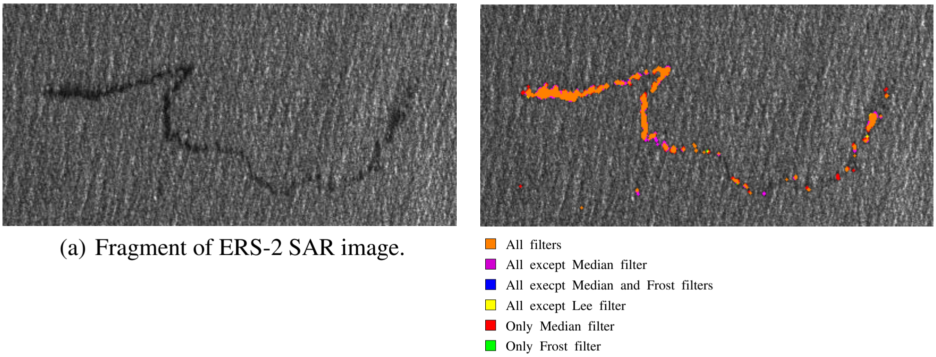
(a) Fragment of Radarsat-1 SAR imagem.

(b) Spot segmented.

Fig. 2. Study area 1

Table 1. Features and its statistics for the scene of image 2(a)

Features	Speckle filter						μ	σ
	Median	Frost	Lee	Kuan	Lee *Enhc.*	Frost *Enhc.*		
A	7.7841	6.5403	5.8833	5.8833	5.8833	5.8833	6.3096	0.3437
P	187.83	142.35	114.48	114.48	114.48	114.48	131.35	13.3407
C	18.99	15.70	13.31	13.31	13.31	13.31	14.655	1.041
S	2.3634857	0.9604389	0.5548226	0.5548226	0.5548226	0.5548226	0.923869	0.323642
OSd	-2.6113989	-1.4395848	-1.064637	-1.064637	-1.064637	-1.064637	-1.38492	0.276952
BSd	14.465359	14.325734	14.275665	14.275665	14.275665	14.275665	14.31562	0.34005
$ConMax$	17.65205	17.660125	17.666537	17.666537	17.666537	17.666537	17.66305	0.0026
$ConMe$	17.640732	17.62996	17.634743	17.634743	17.634743	17.634743	17.63494	0.001529
$GMax$	27.824726	26.774471	24.972369	24.972369	24.972369	24.972369	25.74811	0.557596
GMe	20.501584	18.760858	17.483807	17.483807	17.483807	17.483807	18.19961	0.553663
GSd	19.459883	18.139937	16.709639	16.709639	16.709639	16.709639	17.40639	0.517562



(a) Fragment of ERS-2 SAR image.

(b) Spot segmented.

Fig. 3. Study area 2

enclosed a minor spurius area than the Median filter. This is important for the correct attainment of some desired features. In the same way, the Median filter provided the segmentation of areas not observed for the adaptive filters, mainly in the edge regions. However all the desired extracted characteristics were obtained. The table 1 shows a comparation among all speckle filters tested, which (μ) is the average and (σ) is the standard deviation. The similar values presented for features show that the proposed system is robust.

For ERS-2 image the presented results seem to be good, same that a small part of the spot (where the threshold of decision between background and object is very similar) has not been correctly labeled. This was expected, because for a human operator, this task is even considered difficult. The two analysed scenes posses a high complexity degree for the segmentation stage because of the strong speckle noise.

4 Conclusions and Future Improvements

A segmentation and features extraction process of oil slicks in SAR images was presented. The results for two scenes were satisfactory, providing the computation of all the desired features. The system presented has a high automation level, speeding the process as a whole.

The spekle filters applied in the process of speckle noise filtering present good results. The similarity gotten in the segmentation of the spot for the diverse speckle filters tested shows that the proposed system is robust. It had been observed by the features values obtained. Further, model tests must be completed with more image analysis, also with Envisat image.

With respect to the other parts of the system, the construction of automatic and intelligent tools to spots' selection scene instead of a human operator would become the process totally automatic.

In the classfier block, the research on neural classifiers, as proposed by [1][6][7], already had shown good results. However, other types of learning machines can be tested, providing an automatized system composed of more intelligent tools.

References

1. F. Del Frate and A. Petrocchi and J. Lichtenegger and G. Calabresi, "Neural networks for oil spill detection using ERS-SAR data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, number 5, pp. 2282–2287, Sept. 2000.
2. A.H.S. Solberg and G. Storvik and R. Solberg and V. Volden, "Automatic detection of oil spills in ERS SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, number 4, pp. 1916–1924, July. 1999.
3. A.H.S. Solberg and S.T. Dokken and R. Solberg, "Automatic detection of oil spills in ENVISAT, Radarsat and ERS SAR images," *IEEE International Geoscience and Remote Sensing Symposium, IGARSS '03*, July. 2003, volume 4, pp. 2747–2749.
4. A.H.S. Solberg and R. Solberg, "A large-scale evaluation of features for automatic detection of oil spills in ERS SAR images," *IEEE International Geoscience and Remote Sensing Symposium, IGARSS '96*, May. 1996, volume 3, pp. 1484–1486.
5. A.H.S. Solberg and E. Volden, "Incorporation of prior knowledge in automatic classification of oil spills in ERS SAR images," *IEEE International Geoscience and Remote Sensing Symposium, IGARSS '97*, August. 1997, volume 1, pp. 157–159.
6. G. Calabresi and F. Del Frate and J. Lichtenegger and A. Petrocchi and P. Trivero, "Neural networks for the oil spill detection using ERS-SAR data," *IEEE International Geoscience and Remote Sensing Symposium, IGARSS '99*, June–July. 1999, volume 1, pp. 215–217.
7. F. Del Frate and L. Salvatori, "Oil spill detection by means of neural networks algorithms: a sensitivity analysis," *IEEE International Geoscience and Remote Sensing Symposium, IGARSS '04*, Sept. 2004, volume 2, pp. 1370–1373.
8. R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company,Inc., 1992.
9. Zhenghao Shi and K.B. Fung, "A comparision of Digital Speckle Filters," *IEEE International Geoscience and Remote Sensing Symposium, IGARSS '94*, August. 1994, volume 4, pp.2129–2133.
10. A. Lopes and R. Touzi and E. Nezry, "Adaptive speckle filters and scene heterogeneity," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 28, number 6, pp. 992–1000, Nov. 1990.
11. S. Haykin, *Neural Networks: A Comprehensive Foundation*, International Edition/Second Edition, Prantice Hall,Inc., 1999.

Feature Selection for Neural Networks Through Binomial Regression

Gecynalda Soares S. Gomes and Teresa B. Ludermir

Center of Informatics, Federal University of Pernambuco
Av. Prof. Luis Freire, s/n, Cidade Universitária Recife/PE, 50732-970, Brasil
gecynalda@yahoo.com, tbl@cin.ufpe.br

Abstract. Artificial neural networks have been an interesting alternative to use instead of classic statistical techniques, however, artificial neural networks have some disadvantages, as for example: the training process is long, the choice of topology and input variables (attributes) are difficult. This work uses three models of binomial regression (each model has a different link function) for selecting statistical significant variables for being used as input nodes on each neural network. Hybrid models were constructed, in this paper, in two steps.

Keywords: Link function, hybrid model, binomial regression model, artificial neural network, heart disease, frogs data.

1 Introduction

The use of regression models is far common in practical situations. Through them, we can verify the statistical situation between a response variable (dependent) and one or more predictor variables. The binomial regression model is a special case of the Generalized Linear Models (GLM). In general, this model is used for solving binary classification problems. The structure of a GLM, basically, consist in three parts: (1) one random component made by a variable y with n independent observations, one average vector and one exponential distribution; (2) one systematic component composed by p variables $x_1, \ldots, x_p$, that produce a linear predictor $\eta = X\beta$; and (3) one differentiable monotonic function, known as the link function, that relates both the random and systematic components. More details about these components may be obtained in [7].

The link function must be according to the distribution proposed for the data, and the choice must be done with aims to ease the interpretation of the model. That function relates the linear predictor to the desired value. The more used link functions, in the binomial model, are the logistic, the probit and the log-log complement.

ANN's have been an interesting alternative to the use of classic statistical techniques, for example, the logistic regression, mainly in situations where are complex dependent and independent variables with non-linear relations between them [6]. The backpropagation algorithm often performs the learning of these networks. However, the learning process is lengthy for obtaining the optimal topology of the network, due to the difficulty of identifying the potentially important variables that serves as input nodes for the network. It is known that ANNs using backpropagation do not have a satisfactory performance when the learning process uses examples with many variables. Even under a

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 737–745, 2006.

statistical point of view, examples with noise and many irrelevant variables provide little information. In general, the algorithms remain confused when there are many variables and builds classifiers with low utility [3].

The objective of this paper is to develop a hybrid model using ANN and the binomial regression model, in other words, assembling a neural network model with variables selected from a binomial regression model. Three different models of binomial regression models are used. Each model has a different link function, for selecting statistically significant variables for being used as input nodes for the artificial neural network. The idea of the hybrid model in two stages is based on the paper of Lee and Chen [10], but they used the multivariate adaptative regression splines model (MARS) for selecting variables statistically different. In experiments, to validate the method, are performed two classification problems: (1) heart diseases found [9] and (2) the distribution of the Southern Corroboree frog [2].

The present paper is organized as follows. In Section 2, it is present the binomial regression model and its link functions. The hybrid model in two stages is presented in Section 3. In Section 5 is presented the performed experiments and discussion. The methodology of the experiments is presented in Section 4. Section 6 contains the final remarks.

2 Binomial Regression Model

It is defined a regression model where the dependent variable (Y^*) is the proportion of successes in n independent essays, each one with the probability of occurrence π. Thus, it is assumed that nY^* assumes binomial distribution with index n and parameter π, i.e, $nY^* \sim B(n, \pi)$. The density of Y^* is expressed in $f(y^*, \pi) = \binom{n}{ny^*}\pi^{ny^*}(1-\pi)^{n-ny^*}$, where $0 < \pi, y^* < 1$.

The model is obtained assuming that the average of the Y_t^* may be given by $g(\pi_t) = x_t^\top \beta$ is the linear predictor, $\beta = (\beta_0, \ldots, \beta_p)^\top$ is a vector of unknown regression parameters to be estimated; $x_{t1}, \ldots, x_{tp}$ represent the values of the p co-variables $(p < n)$, which are assumed fixed and known; and $g(\cdot)$ is a monotonic and differentiable function that lifts from the interval $(0, 1)$ to $\mathbb{R}$, denominated link function.For a binomial distribution, in this work it is used three link functions $\eta_1 = \ln\left(\frac{\pi}{1-\pi}\right)$, $\eta_2 = \Phi^{-1}(\pi)$ and $\eta_3 = \ln[-\ln(1 - \pi)]$ are link functions logit, probit and complementary log-log, respectively. The $\Phi(\cdot\,)$ is the accumulated distribution of the standard normal.

3 Hybrid Model

A neural network Multilayer Perceptron (MLP) with an only hidden layer using the training algorithm backpropagation will be adapted to construct the hybrid model in two stages. The input layer of the hybrid model has the significant independent variables obtained from the binomial model. To construct the hybrid model it is followed the next steps: (1) One binomial regression model is adjusted with some determined link function; (2) The statistically significant variables of the model are selected; (3) An artificial neural network model is constructed where its input nodes are the selected variables from the binomial regression model.

4 Methodology

The data described in this section will be used as training and testing sets of the binomial regression, MARS, the neural network models and hybrid models.

The MARS model [5] was used only to feature selection and thus to construct the model hib_MARS with intention to compare with the new hybrid models proposed.

Table 1. Description of the 14 variables used in the model

Variable	Description
1. Age	age in years;
2. Sex	0 = male, 1 = female;
3. Chest pain type	0 = asymptomatic, 1 = typical angina, 2 = atypical angina, 3 = non-anginal pain;
4. Resting blood pressure	in mm Hg;
5. Serum cholestoral	in mg/dl;
6. Fasting blood sugar > 120 mg/dl	0 = no, 1 = yes;
7. Electrocardiographic	0 = normal, 1 = having ST-T wave abnormality, 2 = left ventricular hypertrophy;
8. Maximum heart achieved	rate achieved;
9. Exercise induced angina	0 = no, 1 = yes;
10. ST depression	induced by exercise;
11. Slope ST segment	0 = upsloping, 1 = flat, 2 = downsloping;
12. Vessels colored by flourosopy	0 = none, 1 = one colored vessel, 2 = two colored vessels, 3 = three colored vessels;
13. Thallium	0 = normal, 1 = fixed defect, 2 = reversable defect;
14. Diagnosis	0 = < 50% diameter narrowing, 1 = > 50% diameter narrowing.

The heart data are part of a methodology of experiments called `Proben1`, created for the studying of artificial neural networks [4]. These data were used in the training and testing of the binomial regression and the neural network models, with the intention of predicting heart diseases, based on the reduction of at least one of the four arteries in 50% of the normal, fact that increases the possibility of having and cardiac attack.

Table 2. Description of the 10 variables the frogs data

Variable	**Description**	**Variable**	**Description**
1. Class	0(frogs were absent), 1(frogs were present);	6. NoOfPools	number of potential breeding pools;
2. Northing	reference point;	7. NoOfSites	number of potential breeding sites within a 2 km radius;
3. Easting	reference point;	8. Avrain	mean rainfall for Spring period;
4. Altitude	altitude, in meters1;	9. Meanmin	mean minimum Spring temperature;
5. Distance	distance in meters to nearest extant population;	10. Meanmax	mean maximum Spring temperature.

The dataset consist of 76 variables, but, in this work, just 14 variables were considered as relevant information for predicting diseases in the coronary artery. Table 1 shows the description of the 14 variables, the 14th corresponds to the response variable designated as the diagnostic of the patient. The dataset has 278 patients (patterns), after removing the missing. These data were divided in two sets, having the training set 208 patients and the test set 70 patients. Five different random samples were done to evaluate each model.

The frogs data frame has 212 rows and 10 columns. The data are on the distribution of the Southern Corroboree frog, which occurs in the Snowy Mountains area of New South Wales, Australia. Table 2 presents the description of the 10 variables that we use for modelling the problem. The response variable corresponds to number 1 and the variables from to 2 to 10 correspond to the independent variables. The data set was divided in two sets, 75% of the set for training and 25% for testing. Five different random samples had been made to evaluate each model.

In this article, use a network MLP with 3 layers, the number of hidden nodes is via experiments or trial and error. We, therefore, will also use the trial and error approach with 2, 6, 12 and 18 neurons to determine the appropriate number of hidden nodes for the desired networks. The training of a network is implemented with various learning rates ranging from 0.1 and the 0.0005 and training lengths ranging from 100 to 1,000 iterations until the network converges. The learning of the ANN if gave through the algorithm backpropagation. The function of adopted transference was the logistic sigmoid. The single-layer MLP model will again be adopted in building the hybrid model.

For better understanding of the text it is necessary some definitions, to know, the binomial regression model with link function logit will be called **logit**; the binomial regression model with link function probit will be called **probit**; the binomial regression model with link function complementary log-log will be called **cloglog**; the hybrid model whose input nodes had been selected through the logit model will be called **hyb_log**; the hybrid model whose input nodes had been selected through the probit model will be called **hyb_prob**; the hybrid model whose input nodes had been selected through the cloglog model will be called **hyb_clog** and the hybrid model whose input nodes had been selected through the MARS model will be called **hyb_MARS**.

5 Results

In this Section, the relevant information related to database used is presented. Afterwards the results found in the modeling of the binomial regression, in the learning of the ANNs and the learning of the hybrid model are described.

5.1 Heart Data

The Table 3 shows the results for the selection of the variables through the binomial regression model. For logit model, it can be observed that the statistically significant independent variables at the level of 10% were: sex, chest pain type (angina1, angina2 and angina3), blood pressure (pressure), maximum heart rate achieved, ST segment downsloping (sloping1), colored vessels by fluoroscopy (vessels1, vessels2 and vessels3) and thallium with reversable defect (thallium2).

For cloglog model, the statistically significant independent variables to the level of 10% are the same than the found in the logit model, with the exception of maximum cardiac beating rate, that showed a p-value equals to 0.4352. The results for the feature selection through the probit model not are shown, because the statistically significant independent variables to the level of 10% were the same selected by the logit model. (Table 3)

Table 3. Results for the selection of variables through the binomial regression model

LOGIT MODEL

Variables	Estimate	SE	p-value	Variables	Estimate	SE	p-value	Variables	Estimate	SE	p-value
(Intercept)	−0.34	1.55	0.8221	cholestoral	2.30	1.92	0.2316	sloping1	1.73	0.55	0.0018
age	−1.31	1.30	0.3135	sugar	−0.57	0.62	0.3582	sloping2	0.72	0.95	0.4513
sex	−1.98	0.59	0.0009	electro1	0.85	2.56	0.7392	vessels1	2.30	0.55	0.0000
angina1	−2.37	0.79	0.0027	electro2	0.39	0.42	0.3506	vessels2	3.63	0.84	0.0000
angina2	−1.03	0.60	0.0888	achieved	−3.30	1.78	0.0640	vessels3	2.04	0.98	0.0382
angina3	−2.27	0.56	0.0001	exercise	0.51	0.49	0.2921	thallium1	−0.49	0.88	0.5774
pressure	2.94	1.29	0.0235	depression	1.83	1.54	0.2338	thallium2	1.31	0.47	0.0056

CLOGLOG MODEL

Variables	Estimate	SE	p-value	Variables	Estimate	SE	p-value	Variables	Estimate	SE	p-value
(Intercept)	−1.66	0.88	0.0600	cholestoral	1.01	1.25	0.4155	sloping1	1.01	0.32	0.0019
age	−0.72	0.78	0.3538	sugar	−0.19	0.36	0.5864	sloping2	0.30	0.59	0.6098
sex	−0.97	0.36	0.0070	electro1	0.57	1.33	0.6668	vessels1	1.31	0.31	0.0000
angina1	−1.60	0.52	0.0020	electro2	0.36	0.26	0.1681	vessels2	2.20	0.50	0.0000
angina2	−0.72	0.41	0.0819	achieved	−0.72	0.92	0.4352	vessels3	1.26	0.51	0.0150
angina3	−1.39	0.35	0.0001	exercise	0.25	0.30	0.3901	thallium1	0.10	0.50	0.8353
pressure	1.66	0.81	0.0415	depression	1.53	0.96	0.1102	thallium2	0.94	0.29	0.0011

SE = Standard Error

Table 4. Feature selection results and basis functions of MARS heart data

Variable	Relative importance (%)	Basis function
angina3	100.00	BF1 = max(0, thallium2+2.40E-08)
vessels1	79.57	BF3 = max(0, angina3 +8.07E-10)
vessels2	68.82	BF4 = max(0, depression +5.90E-09)
sloping1	61.36	BF5 = max(0, vessels1 +3.92E-09)
thallium2	59.78	BF6 = max(0, vessels2 +6.26E-10)
angina2	56.37	BF7 = max(0, vessels3 -6.73E-09)
angina1	54.48	BF8 = max(0, sloping1 +2.20E-08)
sex	54.08	BF9 = max(0, sex -1.72E-08)
vessels3	43.14	BF10 = max(0, angina2 -5.85E-10)
depression	25.76	BF11 = max(0, angina1 +5.49E-09)
pressure	8.00	BF12 = max(0, pressure +7.31E-09)

MARS prediction function:

$$Y = 0.722 - 0.195\times\mathrm{BF1}+0.321\times\mathrm{BF3} -0.315\times\mathrm{BF4}-0.268\times\mathrm{BF5}-0.295\times\mathrm{BF6}$$
$$-0.279\times\mathrm{BF7}-0.185\times\mathrm{BF8}+0.178\times\mathrm{BF9} +0.252\times\mathrm{BF10}+0.318\times\mathrm{BF11}-0.262\times\mathrm{BF12}$$

The Table 4 presents the results for feature selection for the heart data with the MARS model, as well as its basic functions for the model. The relative importance of the remaining variable is equal the zero.

In this article, with the purpose to classify the patients in accordance with heart disease, use a network MLP with 3 layers. As input nodes consider the 13 first variables contained in the Table 1, in the output layer consider the last variable of this same table. The presented results to follow come from models of ANN with 6 nodes in the hidden layer and learning rate of the 0.1.

The hybrid models, it is used a MLP network with three layers. As input nodes, the statistically significant independent variables obtained by the binomial regression models were considered; for the output layer, the last variable of Table 1 was chosen. The result presented are from the ANN model with six nodes in the input layer and a learning rate of 0.1.

Through the Table 5, we can observe that, in mean, the model hyb_logit presents the proportion greater of classification (with lesser standard deviation) then candidates to have problems of heart, followed of the hyb_clog model and hyb_MARS, respectively.

For the non-candidates, the hybrid models also are higher to all the other models. We can also observe that the Hyb_MARS model presents greater variability when comparing with the other hybrid models.

Table 5. Results for the test performed by the models for classification patients, candidates or not, to heart disease

Sample	**Model (candidate)**						
	Logit	**Probit**	**Cloglog**	**ANN**	**Hyb_logit**	**Hyb_clog**	**Hyb_MARS**
1	88.5%	88.5%	80.0%	82.8%	91.4%	91.4%	82.8%
2	71.4%	71.4%	71.4%	71.4%	85.7%	82.8%	82.8%
3	88.5%	88.5%	82.8%	88.5%	94.2%	94.2%	94.2%
4	82.8%	88.5%	82.8%	88.5%	94.2%	94.2%	94.2%
5	88.5%	88.5%	82.8%	94.2%	91.4%	91.4%	88.5%
Mean	84.0%	85.1%	80.0%	85.1%	91.4%	90.8%	88.5%
SD	7.4 %	7.6%	4.9%	8.6%	3.5%	4.6%	5.7%

Sample	**Model (non-candidate)**						
	Logit	**Probit**	**Cloglog**	**ANN**	**Hyb_logit**	**Hyb_clog**	**Hyb_MARS**
1	85.7%	85.7%	88.5%	91.4%	91.4%	88.5%	91.4%
2	80.0%	80.0%	82.8%	82.8%	85.7%	85.7%	80.0%
3	85.7%	85.7%	85.7%	82.8%	85.7%	88.5%	88.5%
4	85.7%	85.7%	85.7%	82.8%	85.7%	88.5%	88.5%
5	82.8%	82.8%	88.5%	80.0%	85.7%	82.8%	82.8%
Mean	84.0%	84.0%	86.2%	84.0%	86.8%	86.8%	86.2%
SD	2.5%	2.5%	2.3%	4.3%	2.5%	2.5%	4.6%

SD = Standard Deviation

Table 6 presents the results for the error measures that were used to evaluate the performance of each studied model. The measures found were the sum of the square of errors (SSE) and the mean square error (MSE). The performance for the hybrid and complete ANN models was superior than any other binomial regression model, and the performance of the hybrid model are slightly better than the complete ANN model.

Table 6. Performance results of every evaluated model

	Train				Test			
Model	SSE		MSE		SSE		MSE	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Logit	16.2689	1.7887	0.0783	0.0085	1105.3645	631.7957	15.7852	9.0133
Probit	16.4158	1.7801	0.0789	0.0086	272.3322	176.9368	3.8914	2.5295
Cloglog	15.5135	1.9075	0.0746	0.0092	489.6037	330.9076	6.9886	4.7145
ANN	6.2980	1.2229	0.0303	0.0059	10.6471	2.6537	0.1521	0.0379
Hyb_logit	11.4595	1.7670	0.0551	0.0085	8.7983	2.4129	0.1257	0.0345
Hyb_clog	12.2313	1.6325	0.0588	0.0078	8.7259	2.0840	0.1246	0.0298
Hyb_MARS	11.5792	1.4700	0.0557	0.0070	8.6664	1.8129	0.1238	0.0259

SD = Standard Deviation

5.2 Frogs Data

The results for the feature selection through the probit model not are shown. The statistically significant independent variables to the level of 5% were the same selected by the logit model. This also happens with the results for the selection of the variables through the cloglog model. Table 7 shows the results for the selection of the variables through the logit model. The statistically significant independent variables at the level of 5% were: Distance, NoOfPools and Meanmin.

The Table 8 presents the results for selection of variable of the frogs data through model MARS, as well as its basic functions for the model. The relative importance of the remaining variable is equal the zero.

Table 7. Result for feature selection through of logit model

Variables	Estimate	SE	z value	p-value
(Intercept)	110.49	138.76	0.79	0.4258
Altitude	-55.54	73.36	-0.75	0.4490
Distance	-8.63	3.69	-2.33	0.0194
NoOfPools	6.92	2.15	3.21	0.0012
NoOfSites	0.43	1.06	0.41	0.6807
Avrain	-2.26	11.88	-0.19	0.8492
Meanmin	21.22	6.77	3.13	0.0017
Meanmax	-90.37	80.61	-1.12	0.2622

SE = Standard Error

Table 8. Variable selection results and basis functions of MARS frogs data

Variable	Relative importance (%)	Basis function
meanmin	100.00	BF1 = max(0, distance - 0.944);
altitude	60.43	BF3 = max(0, meanmin - 0.177);
noofpool	46.88	BF4 = max(0, 0.177 - meanmin);
distance	46.58	BF5 = max(0, altitude - 0.189);
meanmax	44.49	BF7 = max(0, noofpool + 0.23e-07);
easting	36.12	BF9 = max(0, 0.100 - meanmax);
northing	20.81	BF11 = max(0, 0.115 - easting);
		BF12 = max(0, northing - 0.657);

MARS prediction function:

$$Y = 1.251 + 7.023 \times \mathrm{BF1} - 1.665 \times \mathrm{BF3} + 17.277 \times \mathrm{BF4} - 124.066 \times \mathrm{BF5} - 0.838 \times \mathrm{BF7} + 90.919 \times \mathrm{BF9} - 7.510 \times \mathrm{BF11} + 5.248 \times \mathrm{BF12}$$

To classify frogs in accordance with its absence or presence in the south of Corroboree, it is used a three layer MLP network. As input nodes, it is considered the seven last variables in Table 2, for the output layer it is considered the first variable of the same table, one hidden layer with two nodes in the hidden layer and a learning rate of 0.0005.

The hybrid models, it is used a MLP network with three layers. As input nodes, the statistically significant independent variables obtained by the binomial regression models were considered; for the output layer, the first variable of Table 2 was chosen. The result presented are from the ANN model with two nodes in the input layer and a learning rate of 0.0005.

Table 9. Result of test by binomial regression model for that classification of absent or present the frogs

Sample	**Model (absent)**						**Model (present)**					
	Logit	**Probit**	**Cloglog**	**ANN**	**Hyb_logit**	**Hyb_MARS**	**Logit**	**Probit**	**Cloglog**	**ANN**	**Hyb_logit**	**Hyb_MARS**
1	78.6%	78.6%	78.6%	85.7%	85.7%	71.4%	79.2%	79.2%	79.2%	75.0%	70.8%	62.5%
2	71.4%	65.7%	77.1%	74.3%	80.0%	62.9%	82.4%	82.4%	76.5%	82.4%	82.4%	76.5%
3	70.6%	67.6%	70.6%	76.5%	73.5%	70.6%	77.8%	77.8%	77.8%	88.9%	94.4%	94.4%
4	75.8%	75.8%	78.8%	87.9%	93.9%	78.8%	68.4%	68.4%	57.9%	47.4%	73.7%	57.9%
5	80.0%	77.1%	82.9%	82.9%	88.6%	82.9%	76.5%	76.5%	76.5%	70.6%	76.5%	58.8%
Mean	75.2%	72.7%	77.6%	81.2%	84.2%	73.3%	76.8%	76.8%	73.7%	72.6%	78.9%	70.0%
SD	4.2%	5.9%	4.4%	5.9%	7.9%	7.8%	5.2%	5.2%	8.8%	15.9%	9.4%	15.6%

SD = Standard Deviation

The Table 9 presents the results of the classification how much the absence or presence of the frogs. For the absence of sapos, we observe that the ANN model is superior to all the binomial regression models, however, this does not happen in the classification of the presence of the frogs. The hybrid model, however, is presented more consistent of the one than the ANN model, or either, the hybrid model is superior to all the other models, as much for absence as for presence of the frogs.

In the case of the of the frogs data, we observe that the Hyb_MARS model presents, with only one exception, the worst results.

Table 10 presents the results for the error measures that were useful to evaluate the performance of each studied model. The measures found were the sum of the square of errors (SSE) and the mean square error (MSE). The performance for the hybrid and complete ANN models was superior than any other binomial regression model, and the performance of the hybrid model, hyb_logit, are slightly better than the complete ANN model.

Table 10. Results of the SSE and MSE values for models

Model	Train				Test			
	SSE		MSE		SSE		MSE	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Logit	23.7992	0.4774	0.1487	0.0030	360.7654	44.5752	6.9379	0.8571
Probit	24.1740	0.3749	0.1511	0.0024	153.3099	16.8466	2.9484	0.3239
Cloglog	23.2167	0.4953	0.1451	0.0031	343.7045	51.0351	6.6095	0.9816
ANN	20.2046	1.0392	0.1262	0.0065	23.0628	1.1270	0.4436	0.0216
Hyb_logit	21.6654	0.5100	0.1354	0.0032	22.0316	0.8181	0.4238	0.0158
Hyb_MARS	9.3521	3.1144	0.0584	0.0195	26.5231	1.8827	0.5102	0.0362

SD = Standard Deviation

6 Conclusion

Through the achieved results it can be noticed that among the binomial regression models, the one that presented a greater average for the correct classification was the model with logit link function, surpassing also the complete ANN model.

In a general, the hybrid model presented the better results. Therefore, to construct a neural network using input feature selection is important and necessary, therefore it improves the results in terms of classification.

The hybrid models whose feature selection was from the binomial regression models had not presented much difference in relation to the model hyb_MARS considered by Lee and Chen (2005) [10]. Therefore, feature selection through the binomial regression model is easier than to construct hybrid models from MARS models because binomial regression models are easier to implement and/or to use.

Moreover, the performance of the hybrid models is superior to the performance of the RNA model whose input nodes have all input variables.

Acknownledgements

The authors thanks the CNPq.

References

1. A. P. Braga, T. B. Ludermir e A. C. P. L. F Carvalho. *Redes Neurais Artificiais - Teoria e Aplicações*, Rio de Janeiro: LTC, 2000.
2. D. Hunter. *The conservation and demography of the southern corroboree frog (Pseudophryne corroboree)*, M.Sc. thesis, Canberra: University of Canberra, 2000.
3. J. A. Baranauskas e M. C. Monard. Metodologias para a seleção de atributos relevantes, *XIII Simpósio Brasileiro de Inteligência Artificial*. SBC: Porto Alegre - Brasil, 1998.
4. L. Prechelt. PROBEN1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules, *Technical report 21/94* Germany: Universitat Karlsruhe, 1994.
5. *MARS 2.0 – for windows 95/98/NT*, Salford Systems, San Diego, CA, 2001.
6. P. Leung e L. T. Tran. Predicting shrimp disease occurrence: artificial neural networks vs. logistic regression, *Aquaculture*, **187**, 35–49, 2000.
7. P. McCullagh e J. A. Nelder. *Generalized Linear Models*, 2nd. Edition. London: Chapman and Hall, 1989.
8. S. Haykin. *Redes Neurais: Princípios e Prática*, 2nd. Edition. Porto Alegre: Bookman, 2001.
9. `http://www.ics.uci.edu/~mlearn/MLRepository.html`,2006.
10. Tian-Shyug Lee e I-Fei Chen. A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines, *Expert Systems with Applications*, **28**, 743–752, 2005.
11. Y. H. Pao. *Adaptive Pattern Recognition and Neural Networks*, 2nd. Edition. New York: Addison-Wesley, 1989, 301p.

Automated Parameter Selection for Support Vector Machine Decision Tree

Gyunghyun Choi* and Suk Joo Bae

Department of Industrial Engineering, Hanyang University,
17 Haengdang-dong Seongdong-gu, Seoul, Korea
ghchoi@hanyang.ac.kr

Abstract. A support vector machine (SVM) provides an optimal separating hyperplane between two classes to be separated. However, the SVM gives only recognition results such as a neural network in a black-box structure. As an alternative, support vector machine decision tree (SVDT) provides useful information on key attributes while taking a number of advantages of the SVM. we propose an automated parameter selection scheme in SVDT to improve efficiency and accuracy in classification problems. Two practical applications confirm that the proposed methods has a potential in improving generalization and classification error in SVDT.

1 Introduction

Pattern recognition has its applications in various fields of practice such as automatic analysis of medical images, quality inspection for automatic manufacturing system, prediction of geological changes, etc. A support vector machine (SVM), which was firstly proposed by Vapnik [4], is based on theoretical structure and it has provided excellent pattern-recognizing achievement in a number of real applications.

In classification problem as an exemplary area of pattern recognition , SVM provides a separating hyperplane between two classes to be separated. Since the separating hyperplane can be applied to various problems, e.g., nonlinear pattern-recognition, function regression, HCI, data mining, web mining, computer vision, artificial intelligence, and medical diagnosis, more active researches on SVM have been done recently.

However, the SVM gives only recognition results such as a neural network in a black-box structure. It hardly provides useful information concerning which attributes affect the results. Accordingly, a support vector machine decision tree (SVDT) was suggested in order to provide information on key attributes while taking a number of advantages of the SVM [2]. The SVDT establishes a mathematical model for each decision nodes and forms a separating hyperplane by solving the model. Determining appropriate parameter values is key issue in the modeling because the separating hyperplane changes according to the parameter values at each decision node, consequently it affects global SVDT performance.

* Corresponding author.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 746–753, 2006.

Bennett [2] searched for proper parameter value by sequently changing the parameters and testing them with a validation set converted from a part of training data. However, it takes much time and efforts to analyze the results and to conduct the test at each decision node. In this paper, we propose an automated scheme for parameter selection in SVDT to resolve such problems.

The paper is organized as follows. In Section 2, we briefly review mathematical models for SVM and SVDT. In Section 3 an automated scheme is proposed to select the parameter in SVDT. In Section 4, we provide two examples to illustrate our procedure. Some concluding remarks are presented in Section 5.

2 Support Vector Machine Decision Tree

2.1 Support Vector Machine

As a tool of pattern recognition, support vector machine (SVM) presents high performance for recognizing a variety of patterns. The SVM like a radial-basis function network linearly projects nonlinear patterns in input space into high-dimensional feature space, and finally linearly analyzes them in the feature space. Via the linear feature space, SVM produces optimal separating hyperplane to resolve classification problems.

For a given dataset $\{(\mathbf{x}_i, t_i), i = 1, \ldots, m\}$, where $\mathbf{x}_i$ is ith training data included in one of two classes, and $t_i \in \{-1, 1\}$ is an indicator representing corresponding class, the SVM searches an optimal separating hyperplane so that it minimizes the distance from the closest support vector to classify every class. For highly overlapped patterns inseparable by linear separating hyperplane, an optimal linear separating hyperplane can be obtained by solving the following optimization problem:

$$\begin{aligned} \min \qquad & \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + \lambda \sum_{i=1}^{m} \eta_i \\ \text{subject to} \qquad & t_i(\boldsymbol{\omega}^T\mathbf{x}_i + b) \geq 1 - \eta_i, \end{aligned} \tag{1}$$

where $\boldsymbol{\omega}$ denotes a vector of distances between separating hyperplanes and the closest support vector, $\eta_i(\geq 0)$ denotes ith slack variable, and λ denotes a penalty parameter for $i = 1, \ldots, m$. Note that all of patterns are perfectly separable when $\eta_i = 0$. The eq. (1) can be solved easily using a Lagrangian dual.

However, it is impossible to discriminate all of patterns with only linear separating hyperplane, thus we need nonlinear separating hyperplane for classifying linearly inseparable patterns. To separate the nonlinear patterns, the SVM nonlinearly projects nonlinear patterns in input space into high-dimensional feature space, and linearly interprets in the feature space. Using a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j)$ for an arbitrary function $\phi(\cdot)$, we can solve classification problem for nonlinear patterns. See [5] for details.

2.2 L_1-Norm Support Vector Machine

Given training data $\{(\mathbf{x}_i, t_i), i = 1, \ldots, m\}$ for $\mathbf{x}_i \in \mathbb{R}^n$, a robust linear programming (RLP) model [1] is defined as

$$\begin{aligned} \min \qquad & \sum_{i=1}^{m} \delta_i \eta_i \\ \text{subject to} \qquad & t_i(\boldsymbol{\omega}^T \mathbf{x}_i + b) \geq 1 - \eta_i, \end{aligned} \tag{2}$$

for $\eta_i \geq 0$, $i = 1, \ldots, m$. Here, m is the number of training data and $\delta_i (> 0)$, representing a misclassification cost for $\mathbf{x}_i$, is defined by

$$\delta_i = \begin{cases} \frac{1}{|c_1|}, & \text{if } \mathbf{x}_i \in c_1 \\ \frac{1}{|c_2|}, & \text{if } \mathbf{x}_i \in c_2. \end{cases} \tag{3}$$

The above form of δ_i guarantees existence of nontrivial solutions [1]. The objective function $\sum_{i=1}^{m} \delta_i \eta_i$ is the degree of permission for $\mathbf{x}_i$ to be closer to an optimal separating hyperplane than a support vector or to be located in the other side of a half space. A separating hyperplane that minimizes classification error is generated by minimizing the objective function.

To introduce a concept of structural risk minimization into the objective function in the RLP, we add L_1-norm $\|\boldsymbol{\omega}\|_1$, then L_1-norm SVM is formulated as

$$\begin{aligned} \min \qquad & \lambda \|\boldsymbol{\omega}\|_1 + (1 - \lambda) \sum_{i=1}^{m} \delta_i \eta_i \\ \text{subject to} \qquad & t_i(\boldsymbol{\omega}^T \mathbf{x}_i + b) \geq 1 - \eta_i, \end{aligned} \tag{4}$$

where λ is a parameter considering trade-off between the margin and error of classification and satisfies $0 \leq \lambda \leq 1$ [3]. By using L_1-norm $\|\boldsymbol{\omega}\|_1$ instead of L_2-norm $\|\boldsymbol{\omega}\|_2$ as in general SVM, the L_1-norm SVM has two advantages:

1. The L_1-norm reduces data dimension more effectively by taking more zero components in $\boldsymbol{\omega}$ than the L_2-norm. The less attributes are, the higher interpretability is .
2. The L_1-norm SVM can use a linear programming instead of a quadratic programming. Widely used linear programming packages such as LINDO™ and CPLEX™ is more efficient and more stable than quadratic programming solvers for large-scale problems, in particular when training data are sparse.

2.3 Support Vector Decision Tree

Nonlinear separating hyperplane is mandatory to solve a variety of pattern classification problems. However, SVM provides only classification results as a blackbox structure, and it fails to support information about key attributes. As an alternative, since a support vector decision tree (SVDT) creates more interpretable rule with fewer attributes, it has a potential in saving costs for data collection by ignoring irrelevant attributes in the analysis later.

In reviewing L_1-norm SVM to generate the SVDT, the L_1-norm SVM generates a linear separating hyperplane that creates two half spaces. L_1-norm SVM is repeatedly applied to each half space, generating two sub-half spaces. We repeat these procedures till some criteria are met, then finally decision trees with nonlinear separating hyperplane. These procedure is called "support vector decision tree (SVDT)" [2].

Unlike a classification and regression trees (CART) and a C4.5, which is classified as a univariate decision tree where a dataset is divided into several meaningful clusters by one attribute, the SVDT is considered as a multivariate decision tree where a dataset is divided into several meaningful clusters by more than one attributes. Potentially, the SVDT achieves better dimension reduction and generates models with low-depth trees in large-scaled dataset, thus it can reduce chances of overfitting and provide more interpretable rules with fewer attributes.

3 Automated Scheme for Parameter Selection in SVDT

Selecting the value of parameter λ in eq. (4) is crucial to execute a decision in SVDT since the global SVDT model is affected by the value. Appropriate value of λ must be selected while considering the trade-off between model generalization and classification accuracy. Bennett [2] used about 1/7 of training data as a validation set and determined the value of λ by testing the generated separating hyperplane with the validation set.

Introducing penalty variable c instead of λ in this paper, the eq. (4) is transformed as

$$\begin{array}{ll} \min & \|\boldsymbol{\omega}\|_1 + c\sum_{i=1}^{m} \delta_i \eta_i \\ \text{subject to} & t_i(\boldsymbol{\omega}^T \mathbf{x}_i + b) \geq 1 - \eta_i, \end{array} \tag{5}$$

and the optimization problem (5) can also be dualized as follows. First, we can define an equivalent problem:

$$\begin{array}{lll} \min & \sum_{j=1}^{n} s_j + c\sum_{i=1}^{m} \delta_i \eta_i & \\ \text{subject to} & t_i(\boldsymbol{\omega}^T \mathbf{x}_i + b) \geq 1 - \eta_i & \forall\ i \\ & \eta_i \geq 0 & \forall\ i \\ & -s_j \leq \omega_j \leq s_j & \forall\ j \end{array}$$

Then, Lagrangian function of the equivalent problem can be defined, moreover the 1^{st} order optimality condition of the Lagrangian function should lead to the following dual problem:

$$\begin{aligned} \max \qquad & Q(\alpha) = \sum_{i=1}^{m} \alpha_i \\ \text{subject to} \qquad & -\mathbf{e} \le \sum_{i=1}^{m} \alpha_i t_i \mathbf{x}_i \le \mathbf{e} \\ & \sum_{i=1}^{m} t_i \alpha_i = 0 \qquad 0 \le \alpha_i \le c, \end{aligned} \tag{6}$$

where $\mathbf{e}$ is $(n \times 1)$ vector in which all of its components are 1. Here, $\sum_{i=1}^{m} \delta_i \eta_i$ is the degree of permission for $\mathbf{x}_i$ to approach a separating hyperplane or to be located in the other side of half-space by passing over the separating hyperplane.

In automated selection for value of the penalty parameter c, as c decreases in eq. (5), which means decrease of penalty on $\sum_{i=1}^{m} \delta_i \eta_i$, $\|\boldsymbol{\omega}\|_1 (\equiv \sum_{j=1}^{n} \omega_j)$ in objective function tends to decreases even if any $\mathbf{x}_i$ is permitted to approach a separating hyperplane or to be included in the other side of half-space (that is, $\eta_i > 0$). As a result, $2/\|\boldsymbol{\omega}\|_2$, margin of separation, increases since $\|\boldsymbol{\omega}\|_2 (\equiv \sum_{j=1}^{n} \omega_j^2)$ increases, and model generalization improves.

On the contrary, as c increases, $\sum_{i=1}^{m} \delta_i \eta_i$ in objective function tends to minimize and $\mathbf{x}_i$ of training data is more likely not to overpass the separating hyperplane. Note that as $\|\boldsymbol{\omega}\|_1$ (related to the margin) and $\sum_{i=1}^{m} \delta_i \eta_i$ (related to classification error) decreases, the margin broadens and classification error minimizes, thus appropriate value of c must be selected by simultaneously considering both margin and classification error.

To deal with in the same scale, $\sum_{i=1}^{n} \omega_i$ and $\sum_{i=1}^{m} \delta_i \eta_i$ are normalized with corresponding standard deviations as

$$\alpha = \frac{\sum_{i=1}^{n} \omega_i}{\sigma_{\sum_{i=1}^{n} \omega_i}}, \qquad \beta = \frac{\sum_{i=1}^{m} \delta_i \eta_i}{\sigma_{\sum_{i=1}^{m} \delta_i \eta_i}},$$

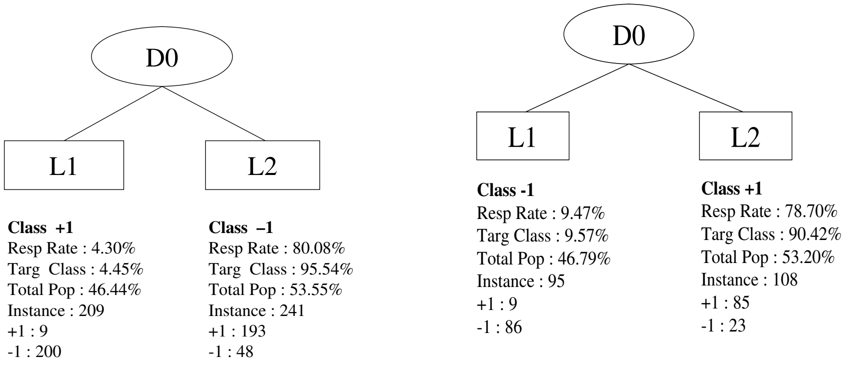
and we search for the c value to minimize α and β simultaneously, equivalently minimize $\alpha + \beta$.

4 Practical Applications

We applied the automated SVDT procedure to *Credit Screening Database* and *Census Income Database* and investigated classification results from the automated scheme for parameter selection. We used SUN Ultra 10 workstation (333 MHz CPU, 512MB Memory) as hardware and AMPL/CPLEX as software to execute the procedure.

4.1 Credit Screening Database

Credit screening database, built up by a Japan credit card company, records 653 customers' information including credit approval results (Granted (+1) or Not Granted (−1)). The customer records consist of 15 attributes; 5 continuous-typed



(a) training data

(b) test data

Fig. 1. Applicative results of the automated SVDT to credit screening example

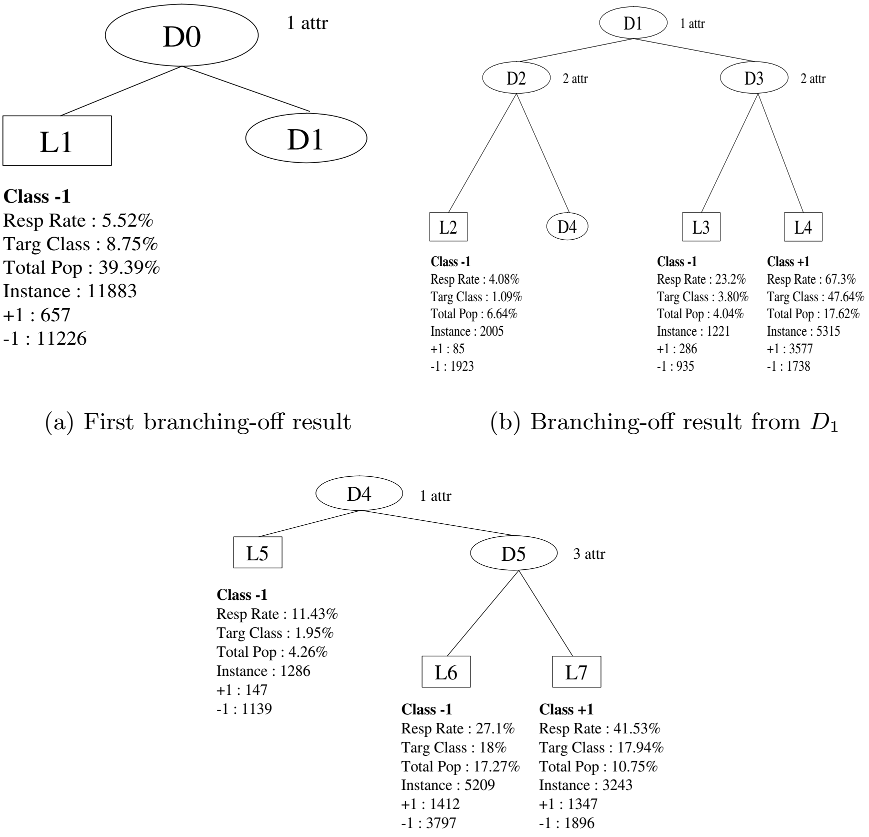
and 9 nominal-typed attributes. The database is sourced from *UCI Machine Learning Repository.* We divided the total data set into 450 (+1: 202, −1: 248) customer records as training data and 203 (+1: 94, −1: 109) as test data. We allocated $1, \ldots, n$ integer values respectively when there are n categories for nominal attributes. Every attribute is normalized with corresponding standard deviation. The objective is to compare classification results from the automated SVDT with real credit approval records.

The applicative results of the automated SVDT to credit screening database is shown in Figure 1-(a). Here, the response rate, defined as the ratio of the number of class +1 at node to the total number of data at corresponding node, is $9/209 = 4.30\%$. The target class is defined as the ratio of the number of class +1 at node to the total number of class +1 in population and its result is $9/202 = 4.45\%$ for $L1$. The total population is defined as the number of instants at corresponding node divided by the number of instants in population.

Finally, the value of c was obtained from the automated procedure as 1.0. As shown in Figure 1-(a), a separating hyperplane to well classify the training data is obtained through only one branching-off. All the attributes except 8th attribute (weight $\omega_8 = -0.998582$ with bias $b = 3.00001$) have zero-weighted values in the analysis, which implies that credit screening data is separable with only one attribute. In confirming performance results for model generalization of the automate SVDT, test result using testing data is given in Figure 1-(b). The proposed method classifies the test data well, connoting better generalization capability.

4.2 Census Income Database

Census income database, established by U.S. census bureau, includes 45,222 demographical information, e.g., age, sex, job, income level, etc. The income level is classified into two groups: less than \$ 50,000 (+1) or larger than \$ 50,000 (+1).



(a) First branching-off result

(b) Branching-off result from D_1

(c) Branching-off result from D_4

Fig. 2. Branching-off results in the automated SVDT: Census income screening training data

The dataset consist of 13 attributes; 6 continuous-typed and 7 nominal-typed attributes. The database is also sourced from *UCI Machine Learning Repository.* We divided the total data set into 32,561 (+1: 7,508, −1: 22,654) as training data and 15,060 (+1: 3,700, −1: 11,360) as test data. We allocated $1, \ldots, n$ integer values respectively when there are n categories for nominal attributes. Every attribute is normalized with corresponding standard deviation. Similarly, the objective is to compare classification results from the automated SVDT with real census income data.

The applicative results of the automated SVDT to census income database is shown in Figure 2. Figure 2-(a) represents the result from first branching-off. The value of c was obtained from the automated procedure as 1.5. Only one attribute ($\omega_6 = -0.780588$ with bias $b = 1.66667$) has non-zero value and the other weight values are found as zeros. The results show that the automated SVDT classifies 40% out of total data as instances having class −1.

The result of branching-off from $L1$ leaf node is given in Figure 2-(b), and branching-off result from only $D4$ leaf node is shown in Figure 2-(c), respectively.

5 Summary and Conclusions

The support vector machine decision tree establishes a mathematical model for each decision nodes and forms a separating hyperplane by solving the model. Determining appropriate parameter values is essential in SVDT. The existing methods suffers from loss of time and efforts to determine the parameter, hence we propose an automated scheme for parameter selection in SVDT. We showed that the proposed method provides efficient classification results with two illustrative examples.

When we select the smaller value than resulting value from the automated scheme, it is likely to generate overfitting and SVDT with high-depth trees. On the contrary, if we select the larger value than resulting value from the automated scheme, it is more likely to generate overfitting as we concentrate on improving classification rate of training data.

In conclusion, parameter value selected from the automated system has a potential in providing more accurate results in the classification problems.

References

1. Bennett, K. P., and Mangasarian, O. L. (1992), " Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", *Optimization Methods and Software*, Vol. **1**, 23–34.
2. Bennett, K. P., Wu, D. H., and Auslender, L. (1998), "On Support Vector Desision Trees for Database Marketing", Rensselaer Polytechnic Institute Math Report No. 98–100, Troy, New York.
3. Bradley, P. S., and Mangasarian, O. L. (1998), " Feature Selection Via Concave Minimization and Support Vector Machines", Mathematical Programming Technical Report, 98-03.
4. Vapnik, V. N. (1998), *Statistical Learning Theory*, Wiley, New York.
5. Scholkopf, B., Burge C. J. C., and Smola, A. J. (1999), *Advanced in Kernel Methods - Support vector Learning*, The MIT Press, New York.

Message-Passing for Inference and Optimization of Real Variables on Sparse Graphs

K.Y. Michael Wong[1], C.H. Yeung[1], and David Saad[2]

[1] Department of Physics, Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong, China
[2] NCRG, Aston University, Birmingham B4 7ET, UK
phkywong@ust.hk, phbill@ust.hk, d.saad@aston.ac.uk

Abstract. The inference and optimization in sparse graphs with *real* variables is studied using methods of statistical mechanics. Efficient distributed algorithms for the resource allocation problem are devised. Numerical simulations show excellent performance and full agreement with the theoretical results.

1 Introduction

Many inference and optimization problems make use of the graphical structures that describe the dependencies between random variables [1]. In contrast to models with extensive inter-dependencies among the variables, the graph-based models can be solved by passing messages between neighbouring variables on the graphs. This message-passing approach has gained recent success in areas such as error-correctig codes [2] and probabilistic inference [3].

Most studies so far have focused on graphs of discrete variables. However, many typical problems involve continuous variables. The main obstacle comes from the need to pass much more complicated messages among the nodes of the graphs, whereas in cases of discrete variables, the messages are countable sets of conditional probability estimates of *discrete values*. There have been attempts to simplify the messages for continuous variables, for example, to parametrize them using eigenfunction decomposition for special cases, but the general feasibility remains an open question [4].

In this paper we study inference and optimization problems on sparse graphs. Based on the analysis, we propose novel message-passing algorithms generally applicable to problems of continuous variables. The method is efficient since the messages consist of only the first and second derivatives of the message functions. The key to the successful simplification is that the messages to a target node are accompanied by information-provision messages from the target node, to first determine the state at which the derivatives should be calculated.

We first consider the general formulation on a sparse graph, and then examine the resource allocation problem, as a vehicle to study the principles and ingredients in message-passing. The problem is interesting for the following reasons. First, it is a well known problem in the area of distributed computing [5] to which

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 754–763, 2006.

significant effort has been dedicated within the computer science community. It is representative of a large class of problems in many other areas where a large number of nodes are required to balance their resources and redistribute tasks, such as reducing internet traffic congestion and streamlining network flows of commodities [6]. Many attempts were made in the computer science community, to find practical heuristic solutions to the distribution of computational load between computers connected by networks.

Second, the problem illustrates the advantages of the message-passing techniques in comparison with the much more computationally demanding global optimization techniques traditionally adopted in this family of problems, such as linear or quadratic programming [7]. For example, the computational complexity of quadratic programming for the load balancing task typically scales as the cube of the system size, whereas capitalizing on the network topology underlying the connectivity of the variables, message-passing scales linearly with the system size. An even more important advantage of message-passing techniques, relevant to practical implementation, is their distributive nature. Since they do not require a global optimizer, they are particularly suitable for distributive control in large or evolving networks.

Third, making use of the conservation of resources on graphs, the problem can be easily transformed to its dual which is exactly solvable using the price iteration scheme. This provides a benchmark for the message-passing method.

In Section 2, we analyze the problem using the Bethe approximation of statistical mechanics. We then present numerical results in Section 3, and derive the new message-passing algorithm on the basis of the analysis in Section 4. The price iteration algorithm is presented in Section 5 for comparison. The study is extended to the unsatisfiable case in Section 6. We conclude the paper in Section 7. Early and partial work in this direction was presented in [8].

2 The Theoretical Framework

We consider a sparse graph with N nodes, labelled $i = 1, \ldots, N$. Each node i is randomly connected to c other nodes. The connectivity matrix is given by $\mathcal{A}_{ij} = 1, 0$ for connected and unconnected node pairs respectively. A link variable y_{ij} is defined on each connected link from j to i. We consider a cost function $E = \sum_{(ij)} \mathcal{A}_{ij}\phi(y_{ij}) + \sum_i \psi(\lambda_i, \{y_{ij}|\mathcal{A}_{ij} = 1\})$, where λ_i is a quenched variable defined on node i. In the context of probabilistic inference, y_{ij} may represent the correlation between observables in nodes j and i, $\phi(y_{ij})$ may correspond to the logarithm of the prior distribution of y_{ij}, and $\psi(\lambda_i, \{y_{ij}|\mathcal{A}_{ij} = 1\})$ the logarithm of the likelihood of the observables λ_i. In the context of resource allocation, $y_{ij} \equiv -y_{ji}$ may represent the current from node j to i, $\phi(y_{ij})$ the transportation cost, and $\psi(\lambda_i, \{y_{ij}|\mathcal{A}_{ij} = 1\})$ the performance cost of the allocation task on node i, dependent on the node capacity λ_i.

We address a generic version of the resource allocation problem, in which N is very large, the capacity λ_i is randomly drawn from a distribution $\rho(\lambda_i)$, and the currents y_{ij} satisfy the link bandwidth constraints $-W \leq y_{ij} \leq W$. For load balancing tasks, $\phi(y)$ is typically a convex function, which will be assumed in our study.

For sufficiently large W and capacity distributions with non-negative average λ, there exist solutions which satisfy the capacity constraints $\sum_j \mathcal{A}_{ij} y_{ij} + \lambda_i \geq 0$. Hence we consider $\psi(\lambda_i, \{y_{ij} | \mathcal{A}_{ij} = 1\}) = \ln[\Theta(\sum_j \mathcal{A}_{ij} y_{ij} + \lambda_i) + \epsilon]$, where $\epsilon \to 0$, and the Θ function returns 1 for a non-negative argument and 0 otherwise. The problem reduces to the load balancing task of minimizing the cost $E = \sum_{(ij)} \mathcal{A}_{ij} \phi(y_{ij})$, subject to the capacity constraints. We call this the *satisfiable* case, which will be considered in Sections 3 to 5 for unconstrained links ($W = \infty$) and $\langle \lambda \rangle > 0$. The unsatisfiable case will be considered in Section 6.

The analysis of the network is done by introducing the free energy $F = -T \ln \mathcal{Z}_y$ for a temperature $T \equiv \beta^{-1}$, where $\mathcal{Z}_y$ is the partition function

$$\mathcal{Z}_y = \prod_{(ij)} \int_{-W}^{W} dy_{ij} \prod_i \Theta \left(\sum_j \mathcal{A}_{ij} y_{ij} + \lambda_i \right) \exp \left[-\beta \sum_{(ij)} \mathcal{A}_{ij} \phi(y_{ij}) \right] . \qquad (1)$$

When the connectivity c is low, the probability of finding a loop of finite length on the graph is low, and the Bethe approximation well describes the local environment of a node. In the approximation, a node is connected to c branches in a tree structure, and the correlations among the branches of the tree are neglected. In each branch, nodes are arranged in generations. A node is connected to an ancestor node of the previous generation, and another $c-1$ descendent nodes of the next generation. Thus, the node is the *vertex* of the tree structure formed by its descendents.

Consider a vertex $V(\mathbf{T})$ of a tree $\mathbf{T}$ having a capacity $\lambda_{V(\mathbf{T})}$, and a current y is drawn from the vertex by its ancestor. One can write an expression for the free energy $F(y|\mathbf{T})$ as a function of the free energies $F(y_k|\mathbf{T}_k)$ of its descendants, that branch out from this vertex, where $\mathbf{T}_k$ represents the tree terminated at the k^{th} descendent of the vertex. The free energy can be considered as the sum of two parts, $F(y|\mathbf{T}) = N_{\mathbf{T}} F_{\text{av}} + F_V(y|\mathbf{T})$, where $N_{\mathbf{T}}$ is the number of nodes in the tree $\mathbf{T}$, F_{av} is the average free energy per node, and $F_V(y|\mathbf{T})$ is referred to as the *vertex free energy*. Note that when a vertex is added to a tree, there is a change in the free energy due to the added vertex. Since the number of nodes increases by 1, the vertex free energy is obtained by subtracting the free energy change by the average free energy. This allows us to obtain the recursion relation

$$\begin{aligned} F_V(y|\mathbf{T}) = & -T \ln \left\{ \prod_{k=1}^{c-1} \left(\int_{-W}^{W} dy_k \right) \Theta \left(\sum_{k=1}^{c-1} y_k - y + \lambda_{V(\mathbf{T})} \right) \right. \\ & \left. \times \exp \left[-\beta \sum_{k=1}^{c-1} \left(F_V(y_k|\mathbf{T}_k) + \phi(y_k) \right) \right] \right\} - F_{\text{av}} . \qquad (2) \end{aligned}$$

For optimization, we take the zero temperature limit of Eq. (2), in which the free energy reduces to the minimum cost, yielding

$$F_V(y|\mathbf{T}) = \min_{\{y_k | \sum_{k=1}^{c-1} y_k - y + \lambda_{V(\mathbf{T})} \geq 0\}} \left[\sum_{k=1}^{c-1} \left(F_V(y_k|\mathbf{T}_k) + \phi(y_k) \right) \right] - F_{\mathrm{av}}. \tag{3}$$

These iterative equations can be directly linked to those obtained from a principled Bayesian approximation, where the logarithms of the messages passed between nodes are proportional to the vertex free energies.

The current distribution and the average cost per link can be derived by integrating the current y' in a link from one vertex to another, fed by the trees $\mathbf{T}_1$ and $\mathbf{T}_2$, respectively; the obtained expressions are $P(y) = \langle \delta(y - y') \rangle_\star$ and $\langle \phi \rangle = \langle \phi(y') \rangle_\star$ where

$$\langle \bullet \rangle_\star = \left\langle \frac{\int dy' \exp\left[-\beta \left(F_V(y'|\mathbf{T}_1) + F_V(-y'|\mathbf{T}_2) + \phi(y') \right)\right] (\bullet)}{\int dy' \exp\left[-\beta \left(F_V(y'|\mathbf{T}_1) + F_V(-y'|\mathbf{T}_2) + \phi(y') \right)\right]} \right\rangle_\lambda . \tag{4}$$

Before closing this section, we mention the alternative analysis of the problem using the replica method [9], which was successfully applied in the physics of disordered systems. The derivation is rather involved (details will be provided elsewhere), but gives rise to the same recursive equation Eq. (2) as in the Bethe approximation.

3 Numerical Solution

The Bethe approximation provides a theoretical tool to analyze the properties of optimized networks. The solution of Eq. (3) is free from finite size effects inherent in Monte Carlo simulations, and can be obtained numerically. Since the vertex free energy of a node depends on its own capacity and the disordered configuration of its descendants, we generate 1000 nodes at each iteration of Eq. (3), with capacities randomly drawn from the distribution $\rho(\lambda)$, each being fed by $c-1$ nodes randomly drawn from the previous iteration. We have discretized the vertex free energies $F_V(y|\mathbf{T})$ function into a vector, whose i^{th} component takes the value $F_V(y_i|\mathbf{T})$.

To compute the average cost, we randomly draw 2 nodes, compute the optimal current flowing between them, and repeat the process 1000 times to obtain the average. Figure 1(a) shows the results as a function of iteration step t, for a Gaussian capacity distribution $\rho(\lambda)$ with variance 1 and average $\langle \lambda \rangle$. Each iteration corresponds to adding one extra generation to the tree structure, such that the iterative process corresponds to approximating the network by an increasingly extensive tree. We observe that after an initial rise with iteration steps, the average energies converge to steady-state values, at a rate which increases with the average capacity.

To study the convergence rate of the iterations, we fit the average cost at iteration step t using $\langle E(t) - E(\infty) \rangle \sim \exp(-\gamma t)$ in the asymptotic regime.

As shown in the inset of Fig. 1(a), the relaxation rate γ increases with the average capacity. It is interesting to note that a cusp exists at the average capacity of about 0.45. Below that value, convergence of the iteration is slow, since the average cost curve starts to develop a plateau before the final convergence. On the other hand, the plateau disappears and the convergence is fast above the cusp. The slowdown of convergence below the cusp is probably due to the appearance of increasingly large clusters of saturated nodes on the network, since clusters of nodes with negative capacities become increasingly extensive, and need to draw currents from increasingly extensive regions of nodes with excess capacities to satisfy the demand.

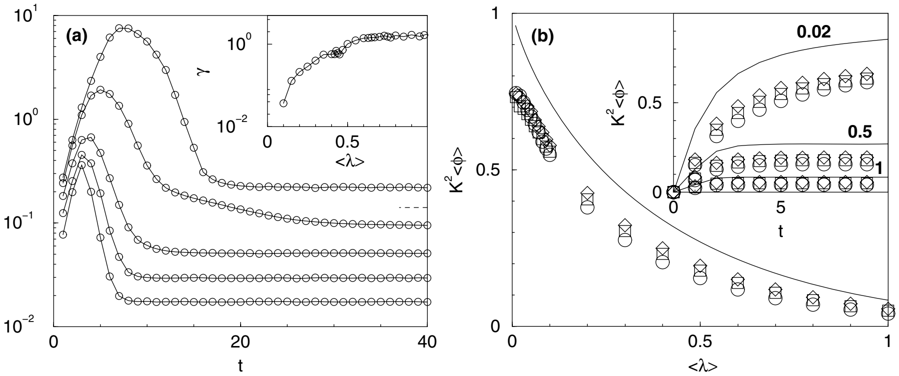


Fig. 1. Results for $N=1000$, $\phi(y) = y^2/2$ and $W = \infty$. (a) $\langle\phi\rangle$ obtained by iterating Eq. (2) as a function of t for $\langle\lambda\rangle = 0.1, 0.2, 0.4, 0.6, 0.8$ (top to bottom), $c=3$ and 200-800 samples. Dashed line: the asymptotic $\langle\phi\rangle$ for $\langle\lambda\rangle = 0.1$. Inset: γ as a function of $\langle\lambda\rangle$. (b) $K^2\langle\phi\rangle$ as a function of $\langle\lambda\rangle$ for $c = 3$ (◯), 4 (□), 5 (◊) and 1000 samples. Line: large K. Inset: $K^2\langle\phi\rangle$ as a function of time for random sequential update of Eqs. (5-6). Symbols: as in (b) for $\langle\lambda\rangle = 0.02, 0.1, 0.5$ (top to bottom).

4 The Message-Passing Algorithm

The local nature of the recursion relation Eq. (3) points to the possibility that the network optimization can be solved by local iterative approaches. However, in contrast to other message-passing algorithms which pass conditional probability estimates of *discrete variables* to neighboring nodes, the messages in the present context are more complex, since they are *functions* $F_V(y|\mathbf{T})$ of the current y. We simplify the message to 2 parameters, namely, the first and second derivatives of the vertex free energies. For the quadratic load balancing task, it can be shown that a self-consistent solution of the recursion relation, Eq. (3), consists of vertex free energies which are piecewise quadratic with continuous slopes. This makes the 2-parameter message a very precise approximation.

Let $(A_{ij}, B_{ij}) \equiv (\partial F_V(y_{ij}|\mathbf{T}_j)/\partial y_{ij}, \partial^2 F_V(y_{ij}|\mathbf{T}_j)/\partial y_{ij}^2)$ be the message passed from node j to i; using Eq.(3), the recursion relation of the messages become

$$A_{ij} \leftarrow -\mu_{ij}, \quad B_{ij} \leftarrow \Theta(-\mu_{ij}) \left[\sum_{k \neq i} \mathcal{A}_{jk} (\phi''_{jk} + B_{jk})^{-1} \right]^{-1},$$

$$\mu_{ij} = \min \left[\frac{\sum_{k\neq i} \mathcal{A}_{jk} [y_{jk} - (\phi'_{jk} + A_{jk})(\phi''_{jk} + B_{jk})^{-1}] - y_{ij} + \lambda_j}{\sum_{k \neq i} \mathcal{A}_{jk} (\phi''_{jk} + B_{jk})^{-1}}, 0 \right], \quad (5)$$

with ϕ'_{jk} and ϕ''_{jk} representing the first and second derivatives of $\phi(y)$ at $y = y_{jk}$ respectively. The forward passing of the message from node j to i is followed by a backward message from node j to k for updating the currents y_{jk} according to

$$y_{jk} \leftarrow y_{jk} - \frac{\phi'_{jk} + A_{jk} + \mu_{ij}}{\phi''_{jk} + B_{jk}}. \quad (6)$$

We note that Eqs. (5-6) differ from conventional message-passing algorithms in that backward messages of the currents are present. As a consequence of representing the messages by the first and second derivatives, the backward messages serve to inform the descendent nodes of the particular arguments they should use in calculating the derivatives for sending the next messages. Furthermore, the criterion that $y_{ij} = -y_{ji}$ provides a check for the convergence of the algorithm.

The message-passing equations further enable us to study the properties of the optimized networks in the limit of large $K \equiv c - 1$, and hence consider the convergence to this limit when the connectivity increases. Given an arbitrary cost function ϕ with nonvanishing second derivatives for all arguments, Eq. (3) converges in the large K limit to the steady-state results $A_{ij} = \max([\sum_{k\neq i} \mathcal{A}_{jk} A_{jk} - \lambda_j]/K, 0)$, $B_{ij} \sim K^{-1}$. Then, $\sum_{k \neq i} \mathcal{A}_{jk} A_{jk}$ becomes self-averaging and equal to $K m_A$, where $m_A \sim K^{-1}$ is the mean of the messages A_{ij} given by $K m_A = \langle \Theta(x - \lambda)(x - \lambda) \rangle_\lambda$. Thus, $y_{ij} \sim \mu_i \sim K^{-1}$. The physical picture of this scaling behavior is that the current drawn by a node is shared among the K descendent nodes. After rescaling, quantities such as $K^2\langle\phi\rangle$, $P(Ky)/K$ and $P(K\mu)/K$ become purely dependent on the capacity distribution $\rho(\lambda)$.

For increasing finite values of K, Fig. 1(b) shows the common trend of $K^2\langle\phi\rangle$ decreasing with $\langle\lambda\rangle$ exponentially, and gradually approaching the large K limit. The scaling property extends to the optimization dynamics (Fig. 1(b) inset). As shown in Fig. 2(a), the current distribution $P(Ky)/K$ consists of a delta function component at $y = 0$ and a continuous component, whose breadth decreases with $\langle\lambda\rangle$. Remarkably, the distributions for different connectivities collapse almost perfectly after the currents are rescaled by K^{-1}, with a very mild dependence on K and gradually approaching the large K limit. As shown in the inset of Fig. 2(a), the fraction of idle links increases with $\langle\lambda\rangle$. The fraction has a weak dependence on the connectivity, confirming the almost universal distributions rescaled for different K.

Since the current on a link scales as K^{-1}, the allocated resource of a node should have a weak dependence on the connectivity. Defining the resource at

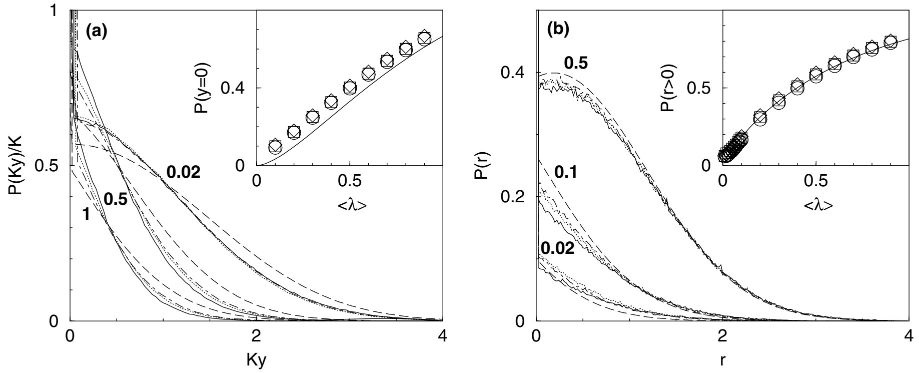


Fig. 2. Results for $N = 1000$, $\phi(y) = y^2/2$, $W = \infty$ and 1000 samples. (a) The current distribution $P(Ky)/K$ for $\langle \Lambda \rangle = 0.02, 0.5, 1$, and $c = 3$ (solid lines), 4 (dotted lines), 5 (dot-dashed lines), large K (long dashed lines). Inset: $P(y=0)$ as a function of $\langle \lambda \rangle$ for $c = 3$ (◯), 4 (□), 5 (◊), large K (line). (b) The resource distribution $P(r)$ for $\langle \lambda \rangle = 0.02, 0.1, 0.5$, large K. Symbols: as in (a). Inset: $P(r > 0)$ as a function of $\langle \lambda \rangle$. Symbols: as in the inset of (a).

node i by $r_i \equiv \sum_j \mathcal{A}_{ij} y_{ij} + \lambda_i$, the resource distribution $P(r)$ shown in Fig. 2(b) confirms this behavior even at low connectivities. The fraction of nodes with unsaturated capacity constraints increases with the average capacity, and is weakly dependent on the connectivity (Fig. 2(b) inset). Hence the saturated nodes form a percolating cluster at a low average capacity, and breaks into isolated clusters at a high average capacity. It is interesting to note that at the average capacity of 0.45, below which a plateau starts to develop in the relaxation rate of the recursion relation, Eq. (3), the fraction of saturated nodes is about 0.47, close to the theoretical percolation threshold of 0.5 for $c = 3$.

5 The Price Iteration Algorithm

An alternative distributed algorithm can be obtained by iterating the chemical potentials of the node. Introducing Lagrange multipliers μ_i for the capacity constraints we get, for links with unlimited bandwidths, $L = \sum_{(ij} \mathcal{A}_{ij} \phi(y_{ij}) + \sum_i (\sum_j \mathcal{A}_{ij} y_{ij} + \lambda_i)$. The extremum condition yields $y_{ij} = \phi'^{-1}(\mu_j - \mu_i)$, and using the Kühn-Tucker condition, μ_i can be solved in terms of μ_j of its neighbours, namely,

$$\mu_i = \min(g_i^{-1}(0), 0); \quad g_i(x) = \sum_j \mathcal{A}_{ij} \phi'^{-1}(\mu_j - x) + \lambda_i. \tag{7}$$

This provides a local iterative method for the optimization problem. We may interpret this algorithm as a price iteration scheme, by noting that the Lagrangian can be written as $L = \sum_{(ij)} \mathcal{A}_{ij} L_{ij} + \text{constant}$, where $L_{ij} = \phi(y_{ij}) + (\mu_i - \mu_j) y_{ij}$.

Therefore, the problem can be decomposed into independent optimization subproblems, each for a current on a link. μ_i is the storage price at node i, and each subproblem involves balancing the transportation cost on the link, and the storage cost at node i less that at node j, yielding the optimal solution. This provides a pricing scheme for the individual links to optimize, which simultaneously optimize the global performance [10]. Simulations show that it yields excellent agreement with the theory Eq. (3) and message-passing Eqs. (5-6).

6 The Unsatisfiable Case

For links with small bandwidth W, or nodes with negative average capacity, there exist nodes which violate the capacity constraint. In these unsatisfiable cases, it is expedient to relax the constraints and search for optimal solutions which limit the violations. Hence we consider the cost $\psi(\lambda_i, \{y_{ij}|\mathcal{A}_{ij} = 1\}) = \Theta(-\sum_j \mathcal{A}_{ij}y_{ij} - \lambda_i)(\sum_j \mathcal{A}_{ij}y_{ij} + \lambda_i)^2/2$. The message-passing algorithm now becomes

$$\begin{aligned} A_{ij} &\leftarrow -\mu_{ij}, \\ B_{ij} &\leftarrow \left\{1+\sum_{k\neq i}\mathcal{A}_{jk}(\phi''_{jk}+B_{jk})^{-1}\Theta\left[W-\left|y_{jk}-\frac{\phi'_{jk}+A_{jk}+\mu_{ij}}{\phi''_{jk}+B_{jk}}\right|\right]\right\}^{-1} ,\end{aligned}\tag{8}$$

where $\mu_{ij} = \min(g_{ij}^{-1}(0), 0)$, with

$$g_{ij}(x) = \sum_{k\neq i}\mathcal{A}_{jk}\max\left\{-W, \min\left[W, \phi'^{-1}(\mu_{jk}-x)\right]\right\} - y_{ij} + \lambda_j - x, \tag{9}$$

The backward message is given by

$$y_{jk} \leftarrow \max\left[-W, \min\left(W, y_{jk} - \frac{\phi'_{jk}+A_{jk}+\mu_{ij}}{\phi''_{jk}+B_{jk}}\right)\right]. \tag{10}$$

The price iteration algorithm now uses $\mu_i = \min(g_i^{-1}(0), 0)$, where

$$g_i(x) = \sum_j \mathcal{A}_{ij}\max\left\{-W, \min\left[W, \phi'^{-1}(\mu_j - x)\right]\right\} + \lambda_i - x, \tag{11}$$

As shown in Fig. 3(a), the average cost per node $\langle E\rangle/N$ increases rapidly when $\langle\lambda\rangle$ enters the unsatisfiable regime, and the results obtained by the theory, the message-passing and price iteration algorithms show excellent agreement. There are 3 types of links in the network: idle ($|y_{ij}| = 0$), unsaturated ($|y_{ij}| < W$) and saturated ($|y_{ij}| = W$). When $\langle\lambda\rangle$ enters the unsatisfiable regime, the fraction of idle links vanishes rapidly, while that of saturated links increases to a steady level, implying that more resources are transported in the links in response to the networkwide demand on resources (Fig. 3(a) inset).

Figure 3(b) shows the simulation results when W varies. For large values of W, the average cost is effectively constant, since the link bandwidth constraints become irrelevant. On the other hand, when W decreases, the average cost increases rapidly, since the links become increasingly ineffective in allocating resources in the network.

As shown in Fig. 3(b) inset, the fraction of saturated links increases when W decreases. It is interesting to note that the fraction of idle links *increases* when W decreases, contrary to the expectation that more links are involved in resource provision. This can be attributed to what we call a *relay effect*. If the links in the network were unconstrained, nodes with sufficiently large violations would have drawn currents from distant neighbours, causing currents to flow through many intermediate nodes, which act as relays for resource transmission. However, when W is small, the currents drawn by nodes with violations from their nearest neighbours may have already saturated the links, and there is no use to draw currents from further neighbours. In the limit of vanishing W, the links are exclusively either idle or saturated. In this limit, a link is idle only when both nodes at its ends have positive λ. Hence the fraction of idle links is $f_{\rm idle} = 1 - f_{\rm sat} = [P(\lambda > 0)]^2$. Since the transportation cost is negligible in this limit, the contribution to the average cost only comes from the violated nodes, given by $\langle E\rangle/N = \langle \Theta(-\lambda)\lambda^2/2\rangle_\lambda$. These predictions are consistent with the simulation results in Fig. 3(b).

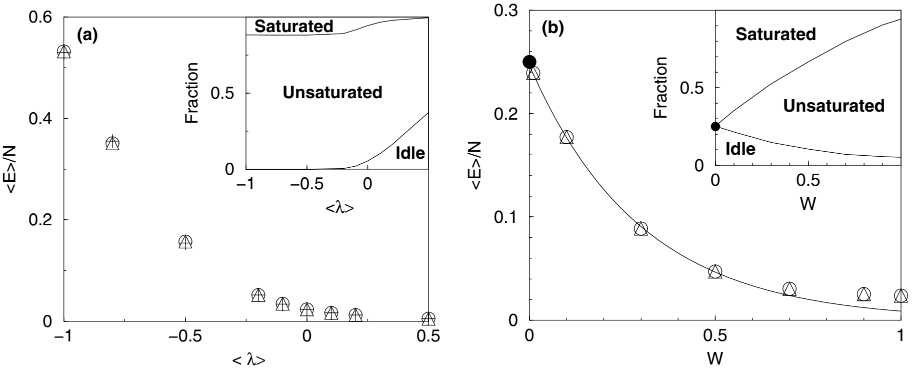


Fig. 3. Results for $N = 1000$, $c = 3$, $\phi = 0.05y^2$ and 100 samples. (a) $\langle E\rangle/N$ as a function of $\langle\lambda\rangle$ for $W = 1$. Symbols: Bethe approximation (+), message-passing (△), price iteration (◯). Inset: the fraction of idle, unsaturated and saturated links as a function of $\langle\lambda\rangle$ for $W = 1$; the vertical height of each region for a given $\langle\lambda\rangle$ corresponds to the respective fraction. (b) $\langle E\rangle/N$ as a function of W for $\langle\lambda\rangle = 0$. Symbols: message-passing (△), price iteration (◯), $W \to 0$ theoretical limit (●). Line: exponential fit for small values of W. Inset: the fraction of idle, unsaturated and saturated links as a function of W for $\langle\lambda\rangle$. Symbol: $W \to 0$ theoretical limit of the fraction of idle links (●).

7 Conclusion

We have studied a prototype problem of resource allocation on sparsely connected graphs. The resultant recursion relation leads to a message-passing algorithm for optimizing the average cost, which significantly reduces the computational complexity of the existing method of global optimization, and is suitable for online distributive control. The suggested 2-parameter approximation produces results with excellent agreement with the original recursion relation and the price iteration algorithm. The Bethe approximation also reveals the scaling properties of this model, showing that the resource distribution on the nodes depends principally on the networkwide availability of resources, and depends only weakly on the connectivity. Links share the task of resource provision, leading to current distributions that are almost universally dependent on the resource availability after rescaling.

While the analysis focused on fixed connectivity and zero temperature for optimization, it can accommodate any connectivity profile and temperature parameter. For instance, we have considered the effects of adding anharmonic terms and frictional terms to the quadratic cost function. The message-passing algorithm can be adapted to these variations, and the results will be presented elsewhere. Besides, it can be used for analyzing a range of inference problems with continuous variables other than optimization. These advances open up a rich area for further investigations with many potential applications in optimization and inference.

Acknowledgments. This work is partially supported by research grants HKUST 6062/02P, DAG04/05.SC25 and DAG05/06.SC36 of the Research Grant Council of Hong Kong and by EVERGROW, IP No. 1935 in the FET, EU FP6 and STIPCO EU FP5 contract HPRN-CT-2002-00319.

References

1. Jordan M. I. (ed.), *Learning in Graphical Models*, MIT Press, Cambridge, MA (1999).
2. Opper M. and Saad D., *Advanced Mean Field Methods*, MIT press (2001).
3. MacKay D.J.C., *Information Theory, Inference and Learning Algorithms*, CUP UK (2003).
4. Skantzos N. S., Castillo I. P. and Hatchett J. P. L., arXiv:cond-mat/0508609 (2005).
5. Peterson L. and Davie B.S., *Computer Networks: A Systems Approach*, Academic Press, San Diego, CA (2000).
6. Shenker S., Clark D., Estrin D. and Herzog S., *ACM Computer Comm. Review* **26** (1996) 19.
7. Bertsekas D., *Linear Network Optimzation*, MIT Press, Cambridge, MA (1991).
8. Wong K. Y. M. and Saad D., arXiv:cond-mat/0509794 (2005).
9. Mézard M., Parisi P. and Virasoro M., *Spin Glass Theory and Beyond*, World Scientific, Singapore (1987).
10. Kelly F. P., *Euro. Trans. on Telecommunications & Related Technologies* **8** (1997) 33.

Analysis and Insights into the Variable Selection Problem

Amir F. Atiya

Dept Computer Engineering Cairo University Giza, Egypt
amir@alumni.caltech.edu

Abstract. In many large applications a large number of input variables is initially available, and a subset selection step is needed to select the best few to be be used in the subsequent classification or regression step. The designer initially screens the inputs for the ones that have good predictive ability and that are not too much correlated with the other selected inputs. In this paper, we study how the predictive ability of the inputs, viewed individually, reflect on the performance of the group (i.e. what are the chances that as a group they perform well). We also study the effect of "irrelevant" inputs. We develop a formula for the distribution of the change in error due to adding an irrelevant input. This can be a useful reference. We also study the role of correlations and their effect on group performance. To study these issues, we first perform a theoretical analysis for the case of linear regression problems. We then follow with an empirical study for nonlinear regression models such as neural networks.

1 Introduction

Variable selection is an important first step in the majority of the machine learning approaches. There are two major approaches for subset selection [3], [5], [8]. In the first one, the so-called *filter* approach, one tests the input variables without considering which classification or regression method is going to be used. This is typically done using general criteria that measure the performance potential of the input variables. The other approach is called the *wrapper* approach [6], and considers testing the inputs in conjunction with the classification or regression method that will be used. Because of the combinatoric nature of the needed search, approximate methods such as the forward sequential selection method or the backward sequential selection method are typically used. In the majority of large scale applications typically a combination of filter/wrapper approaches is performed. The user initially screens the typically numerous available input variables and removes irrelevant and correlated inputs. This step often includes judgement from the application domain, and is based on looking at a number of performance measures such as the individual prediction or classification performance of the input. and the correlation coefficients between the inputs. Once the bad inputs are screened out, a more quantitative wrapper method is implemented to select the best few.

The goal of this paper is to consider this first screening step, and develop insights into the several measures used. An important measure considered is the

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 764–775, 2006.

individual prediction performance of the input. Given that the considered inputs, viewed individually, are by themselves pedictive, what are the chances that as a group they will perform well. This question will be addressed in this research. The other issue is how to determine whether an input is an irrelevant input (or a "noise" input). We develop a formula for the distribution and the expectation of the change in performance by adding an irrelevant input, as this can be a useful reference or a benchmark. It has been a common wisdom that two inputs that are highly correlated should not be selected together. The reason is that the second input does not provide much extra information. We will develop some insights into the role of correlations in predicting how good a subset of inputs works. In all the analysis we consider linear regression problems to develop theoretical results. Generalization of this study to nonlinear regression is the second step of our study and a large scale empirical study is performed for that case. In this paper we do not consider here finite-sample or overfitting effects. We assume that the training set is large enough to have accurate measures of the error.

2 Mathematical Preliminaries

Let N and M denote the number of input variables and training data points respectively. Assume $N \leq M$. Let $x(m) \in \mathcal{R}^{1\times N}$ and $y(m) \in \mathcal{R}$ represent respectively the m^{th} training set input vector and corresponding target output. Let us arrange the input vectors in a matrix $X \in \mathcal{R}^{M\times N}$ with the rows being $x(m)$, and let us also arrange the target outputs $y(m)$ in a column vector y. Also, let the columns of X be denoted as x_n.

In the linear regression problem, the goal is to find the weight vector (vector of regression coefficients) that minimizes the sum of square error:

$$E = \|Xw - y\|^2 \tag{1}$$

The solution is given by

$$w = \left(X^T X\right)^{-1} X^T y \tag{2}$$

and the resulting minimum error is given by

$$E = y^T \mathcal{P}_X^{\perp} y \equiv y^T \left(I - X(X^T X)^{-1} X^T\right) y \tag{3}$$

where the matrix $\mathcal{P}_X^{\perp}$ represents the projection on the null space of X. Another useful quantity is the projection onto the range of X:

$$\mathcal{P}_X y = X(X^T X)^{-1} X^T y \tag{4}$$

Maximizing the length of this projection vector is equivalent to minimizing the error, as we have the fundamental formula (see [7])

$$\|y\|^2 = \|\mathcal{P}_X y\|^2 + \|\mathcal{P}_X^{\perp} y\|^2 \tag{5}$$

(see Figure 1 for an illustration). In many cases in this research it is easier to consider the projection onto the range of X rather than on the null-space of X.

3 Some Insight

To appreciate the complexity of the studied issues of how individual prediction performance and correlations factor in, we cite the following example that shows how often unintuitive conclusions can sometimes be. The problem is given in Figure 1. The vector x_1 is a column vector of X. It represents input no. 1, where the elements of the vector are the values of the specific input for the different training patterns. Similarly x_2 represents input no. 2. The vector y is, as defined last section, the target outputs of the different training data points. For simplicity, we took x_1 and x_2 to lie in the x-y plane, while the vector y is slightly tilted above the x-y plane. The angle between x_1 and y is close to 90 degrees and so input no. 1 has very little individual predictivity. (The reason is that the projection of y on x_1 is a very small vector.) Similarly, x_2 has very little individual predictivity. In addition, x_1 and x_2 are highly correlated as the angle between them is small. It would seem that both inputs would be quite useless: they are not very predictive, and they are highly correlated. The surprising outcome, however, is that using both inputs together leads to very small error. The reason, as one can see graphically, is that the projection vector $\mathcal{P}_X y$ of y onto the x-y plane (the plane containing x_1 and x_2) has a large magnitude. Of course the result in unintuitive, and this prompts us to attempt further analysis and study of these relationships.

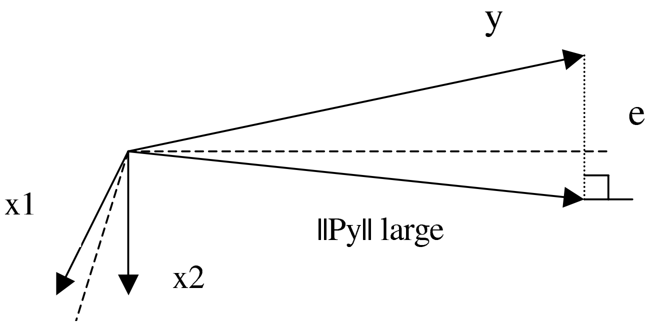


Fig. 1. An illustration of some of the unexpected behavior for the variable selection problem. Inputs x_1 and x_2, viewed individually perform poorly in terms of predicting y, but togther they perform very well.

4 A Study on Irrelevant Inputs

The other issue is how to determine whether an input is an irrelevant input (or a "noise" input). In this section we derive a formula for the probability density of the change in error due to adding an irrelevant input, such as a completely random input. This experiment can yield a useful reference or null hypothesis. If for example in some application an added input does not improve the performance beyond what is expected of a random input, then that considered input is a suspect input. The following is the result.

Theorem: Assume the N input vectors x_n (columns of X) are distributed according to a spherical distribution, i.e. $p(x_n) = \mathrm{fn}(\|x_n\|)$. Also, assume y is distributed according to any spherical distribution, and let $y, x_1, \ldots, x_N$ be independent. Then, the distribution of $\mathcal{E} \equiv E/\|y\|^2$ (normalized error) is given by

$$p(\mathcal{E}) = \mathrm{Beta}\left(\frac{M-N}{2}, \frac{N}{2}\right) \equiv \frac{\Gamma\left(\frac{M}{2}\right)}{\Gamma\left(\frac{M-N}{2}\right)\Gamma\left(\frac{N}{2}\right)} \mathcal{E}^{\frac{M-N}{2}-1}\left(1-\mathcal{E}\right)^{\frac{N}{2}-1} \tag{6}$$

and the expectation is given by

$$\bar{\mathcal{E}} = 1 - \frac{N}{M} \tag{7}$$

Proof: Because, what matters is the direction, not the length, of the vector y, we can normalize the length of each y, so that it lies on a hypersphere, and the density of y is uniform on the hypersphere (because of the spherical distribution assumption). Alternatively, we could assume any arbitrary spherical distribution and work with the unnormalized y. All spherical distributions will give precisely the same result because they are equivalent to the normalized hypersphere case. For that purpose, we assume that y is multivariate Gaussian with mean zero and covariance matrix the identity matrix.

Let L be the linear subspace spanned by $x_1, \ldots, x_N$. Let us rotate the (M-dimensional) space of $x_1, \ldots, x_N, y$ such that the first N components coincide on L. Then, $\mathcal{P}_X y$, i.e. the projection of y on L, is given by

$$\mathcal{P}_X y = (y_1, \quad y_2, \quad \ldots \quad y_N, \quad 0, \quad 0, \quad \ldots \quad 0)^T \tag{8}$$

The normalized error is given by

$$\mathcal{E} = \frac{\|\mathcal{P}_X^{\perp} y\|^2}{\|y\|^2} = \frac{\|y\|^2 - \|\mathcal{P}_X y\|^2}{\|y\|^2} = 1 - \frac{\sum_{i=1}^{N} y_i^2}{\sum_{i=1}^{M} y_i^2} \tag{9}$$

From [1], we know that the ratio of (9) in the RHS for the case of independent Gaussian variables y_i is a beta distribution. Specifically, the normalized error is given by the formula in (6)

It is interesting to see that random inputs reduce the error in a linear fashion. So one should generally be wary if any new input added just a small improvement in performance.

5 The Effect of Individual Prediction Performance

We have seen in Section 3 that a poor individual prediction performance might not make the input useless. However, the converse of the above argument is not valid. An input with good individual prediction performance will guarantee at least that amount of performance for the group. This is a well-known fact because

$$\|\mathcal{P}_{X_S}\| \leq \|\mathcal{P}_X\| \tag{10}$$

where X_S is the matrix of a subset S of columns of X. But what is average case performance? How would the input's individual prediction performance reflect on the group's performance?

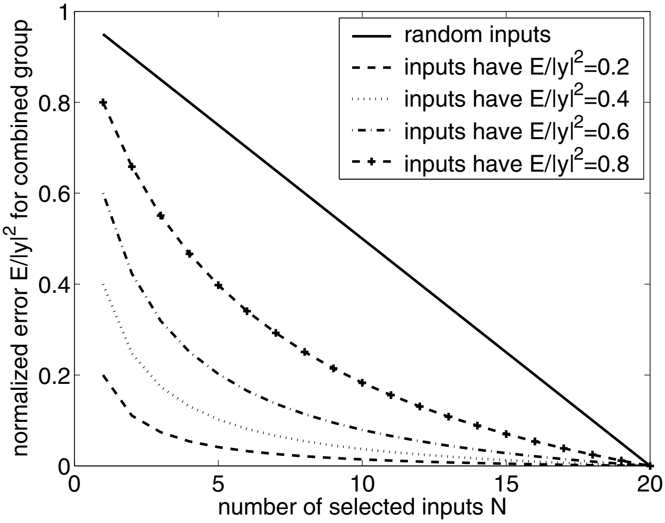


Fig. 2. Input group performance given specific individual input performance. All inputs have the same individual normalized error as given in the legend: 0.2, 0.4, 0.6, or 0.8. Also shown is the case when the inputs are random. The normalized error for the group of inputs is graphed as a function of the number of selected inputs.

We performed the following experiment. We generated the columns x_n of X as well as y from a spherical distribution. For simplicity, we normalize y such that $\|y\| = 1$. We generate the points subject to the condition that $\|\mathcal{P}_{x_n} y\|^2 = q$ (guaranteeing that the error E on each individual input is exactly $1 - q$). Then, we measure the mean of the group performance (performance of the number of inputs together). It was hard to derive the expression for the distribution analytically, so we performed a simulation. Figure 2 shows the expectation of the error E against N for various values of q for $M = 20$. In the same graph we have plotted the expected error curve for random vector case as developed last section. As we can see from the plot, the performance improves as we add more inputs. The outperformance over the random input case is always preserved, thus attesting to the importance of looking at the individual performance of the inputs. There is some saturation effect when N gets larger, but this is expected because it is the early few inputs that bring in most of the information.

6 The Role of Correlations

The role of correlations is more studied in the statistics literature (see [4]), but we will add here some insights. It is well-known that any nonsingular linear transformation of the input vectors will not change the prediction performance of the subsequently designed regression model. Since the linear transformation can arbitrarily adjust the correlation structure among the variables, one could imagine that the existence of high or low correlations among two input variables might not affect the performance of the group of inputs. We have proved the following interesting theorem that supports that argument:

Theorem: Assume the N input vectors x_n (columns of X) are distributed according to a spherical distribution, i.e. $p(x_n) = \mathrm{fn}(\|x_n\|)$. Also, assume that y is distributed according to any spherical distribution. Then, the distribution of the normalized error $\mathcal{E} \equiv E/\|y\|^2$ is independent of the correlation coefficient of any two given inputs i and j.

Proof: Without loss of generality, let $x_1 = (1,\ 0,\ 0, \ldots,\ 0)$. Let S_1 be the set of points x_2 for which $\mathrm{corr}(x_1, x_2) = \rho_1$. It is given by the following formulas

$$x_{21} = \rho_1 \tag{11}$$

$$\sum_{i=2}^{N} x_{2i}^2 = 1 - \rho_1^2 \tag{12}$$

where x_{ni} denotes the i^{th} component of vector x_n. Similarly, let S_2 be the set of points x_2 for which $\mathrm{corr}(x_1, x_2) = \rho_2$, where $\rho_2 < \rho_1$. It will obey similar equations as above with ρ_2 replacing ρ_1.

Consider the following transformation that maps S_1 into S_2:

$$x_2' = \beta x_1 + \gamma(x_2 - x_1) \tag{13}$$

$$x_n' = x_n, \qquad \text{for } n \neq 2 \tag{14}$$

where

$$\beta = \rho_2 + \sqrt{\frac{(1-\rho_2^2)(1-\rho_1)}{(1+\rho_1)}} \tag{15}$$

$$\gamma = \sqrt{\frac{1-\rho_2^2}{1-\rho_1^2}} \tag{16}$$

It can be shown that with this choice of γ and β, $x_{21}' = \rho_2$ and $\sum_{i=2}^{N} {x'_{2i}}^2 = 1 - \rho_2^2$.

So essentially, S_1 and S_2 are two spheres with common center and with S_2 larger and enclosing S_1. The mapping is performed through a ray that is based at the center of the spheres and that projects each point in S_1 to its closest point on

S_2. It can be seen that after the new transformation, the normalized error $\mathcal{E}$ does not change, i.e.

$$\mathcal{E}\Big(x(1), x(2), \ldots, x(m)\Big) = \mathcal{E}\Big(x(1), x'(2), \ldots, x(m)\Big) \tag{17}$$

The reason is that x'_2 is linearly related with x_1 and x_2, and hence the projection space (i.e. the linear space spanned by the x'_n's) is the same (as that of $x_1,\ x_2, \ldots, x_N$). Therefore

$$p\big(\mathcal{E}, \mathrm{corr}(x_1, x_2) = \rho_2\big) = p\big(\mathcal{E}, \mathrm{corr}(x_1, x_2) = \rho_1\big)\gamma^{M-1} \tag{18}$$

where γ^{M-1} is the determinant of the Jacobian of the transformation. Also

$$p\big(\mathrm{corr}(x_1, x_2) = \rho_2\big) = p\big(\mathrm{corr}(x_1, x_2) = \rho_1\big)\gamma^{M-1} \tag{19}$$

where γ^{M-1} here represents also the ratio of the two spheres' surface areas. Hence

$$p\big(\mathcal{E}|\mathrm{corr}(x_1, x_2) = \rho_2\big) = p\big(\mathcal{E}|\mathrm{corr}(x_1, x_2) = \rho_1\big) \tag{20}$$

Hence the density of $\mathcal{E}$ is independent of the correlation between x_1 and x_2.

The harm that highly correlated inputs bring, however, is (see [4]) the possible numerical problems encountered due to having very large weight vector values, and the extra degrees of freedom in the model that increase the complexity of the model without buying us a better performance.

7 Extension to Nonlinear Models

All the previous analysis considers linear problems. The question is whether the same findings apply to nonlinear regression models such as neural networks. It is very hard to derive an analysis similar to the above for nonlinear models. We therefore perform here only an empirical study. For this purpose we have run a large scale experimental simulation.

Like in the case of linear regression models, we do not consider here the overfitting or data insufficiency issue, as this is out of the scope of the paper. Performance in the presence of large enough training set is only considered. This does not mean that the study is not useful to cases of small training sets. Having good performance in the presence of presumably sufficient training data is a baseline performance that has to be secured at first. Once this is achieved the overfitting issue could then be addressed, for example by limiting the complexity of the model.

We consider multilayer networks, and used four the real-world regression problems:

- The CPU data set of the DELVE data repository [2].
- The HOUSE16H data set of the DELVE data repository [2].
- The end-to-end packet loss rate prediction problem of [9].
- The packet round trip time prediction problem of [9].

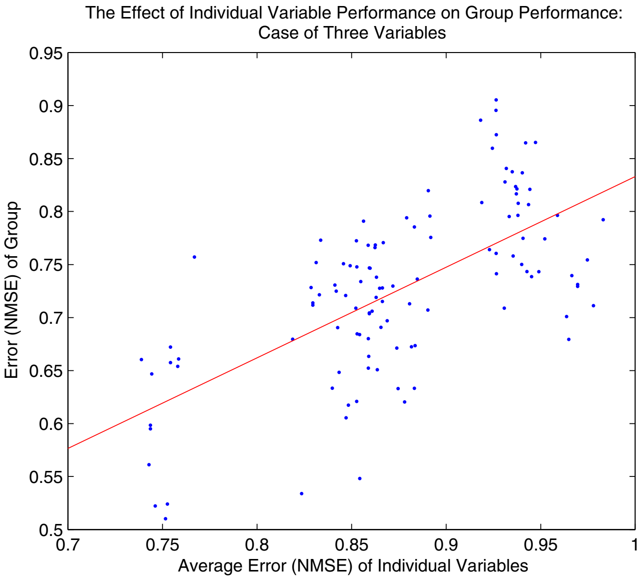


Fig. 3. The relationship between the group performance (NMSEGROUP) and the average individual variable performance (NMSEAVG), for the case of selections of three variables. Each point represents a selection of variables selected at random from among the four benchmark problems considered. Also shown is the regression line for the points.

These are large scale data sets with many input variables: (21, 16 ,51 and 51 in respectively the four data sets). We used 5000 training patterns for all the problems, and considered a ten-hidden node network. We used Levenberg-Marcquardt's training algorithm, and trained every network for 1000 iterations using a learning rate of 0.1.

In the first experiment, we are exploring the effect of individual variable performance on group performance. We trained a multilayer network with each variable as the single only one input, and observed the error. The error measure we used is the normalized mean square error (NMSE), which is defined as the mean square error (MSE) divided by the mean of the square of the target values. So it is a normalized measure of error that generally (but not always) varies from 0 to 1. After we have observed the individual variable performance, we group the variables into groups with approximately similar NMSE. We have three groups, variables with NMSE in the range from 0.7 to 0.8, those with NMSE in the range from 0.8 to 0.9, and those with NMSE in the range from 0.9 to 1. From each group three variables are selected at random. The reason we limit the choice to variables from only one group at a time is to have the selected three variables almost uniform in individual performance (somewhat similar to the experiment in Section 5).

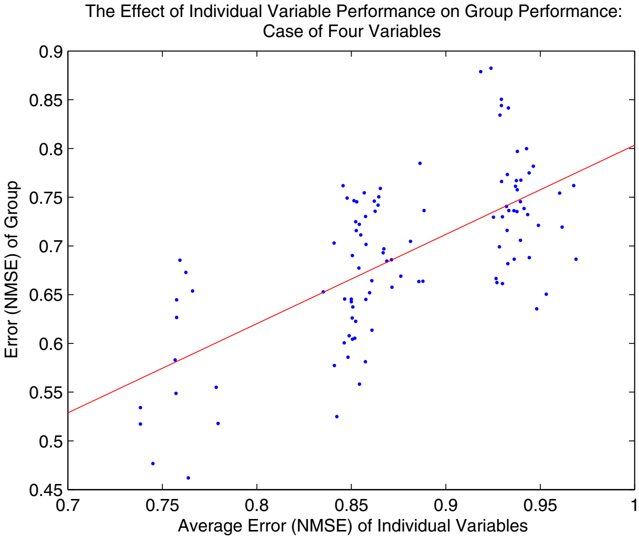


Fig. 4. The relationship between the group performance (NMSEGROUP) and the average individual variable performance (NMSEAVG), for the case of selections of four variables. Each point represents a selection of variables selected at random from among the four benchmark problems considered. Also shown is the regression line for the points.

The average of the individual performances (measured by NMSE) for the selected three variables is computed (call it AVGNMSE). Also, we compute the average pairwise correlation coefficients among the three variables (call it AVGCORR). For example, assume we targeted the group with NMSE from 0.8 to 0.9. We selected three variables at random from this group, and their individual NMSE's turned out to be 0.82, 0.86, and 0.88. Assume that the three inputs' correlation matrix turned out to be:

$$C = \begin{pmatrix} 1 & 0.2 & -0.3 \\ 0.2 & 1 & 0.5 \\ -0.3 & 0.5 & 1 \end{pmatrix} \tag{21}$$

Then, AVGNMSE=(0.82+0.86+0.88)/3 and AVGCORR=(0.2+0.3+0.5)/3. Note that we took the absolute value of the correlation coefficient, because negating one variable (which does not really alter the information content of the variable) will flip the sign of the correlation coefficient. After the three variables are selected, we use these as inputs to a multilayer network. The network is trained and the performance (NMSEGROUP) is observed. The next step is to plot a scatter diagram of the group performance NMSEGROUP against the average individual performance (NMSEAVG). Each point in the scatter diagram represents a selected subset of three variables for any of the four considered benchmarks. Thus all random selections, groups, and benchmark problems are

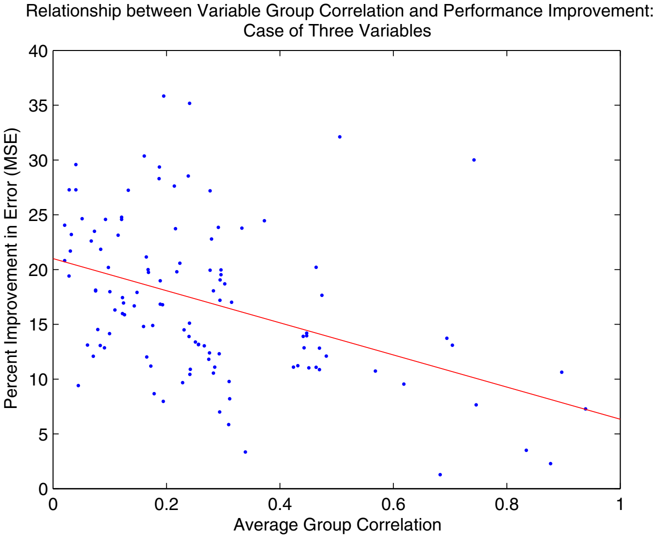


Fig. 5. The relationship between the group performance improvement and the average correlation coefficient among the variables in the selected group, for the case of selections of three variables. Each point represents a selection of variables selected at random from among the four benchmark problems considered. The performance improvement is measured in terms of the percent change in the mean square error. Also shown is the regression line for the points.

represented in this plot. The reason is to get an idea of the general behavior for different classes of problems. Each point in the scatter diagram depicts the group performance NMSEGROUP against NMSEAVG for the considered random selection of three variables. The idea is to see how individual performance carries through to affect group performance. Figure 3 shows this scatter diagram. One can see that there is a strong positive effect between individual performance and group performance. To display such effect, the figure also shows a regression line that approximates such a relationship. It is seen that the regression line is far from flat. We repeated that experiment with everything similar as before except that the selected subset now has four input variables instead of three. Figure 4 shows the scatter diagram for for the four-input case. We see a similar phenomenon as the case of three variables.

In the next experiment we explore the relationship between the average correlation of the inputs within a selection with group performance. We plotted scatter diagrams showing group performance (NMSEGROUP) improvement against the average intra-group correlation (AVGCORR), using the points obtained in the last experiments. The group performance improvement is defined as:

$$\text{Percent improvement} = 100(NMSEAVG - NMSEGROUP)/NMSEAVG \tag{22}$$

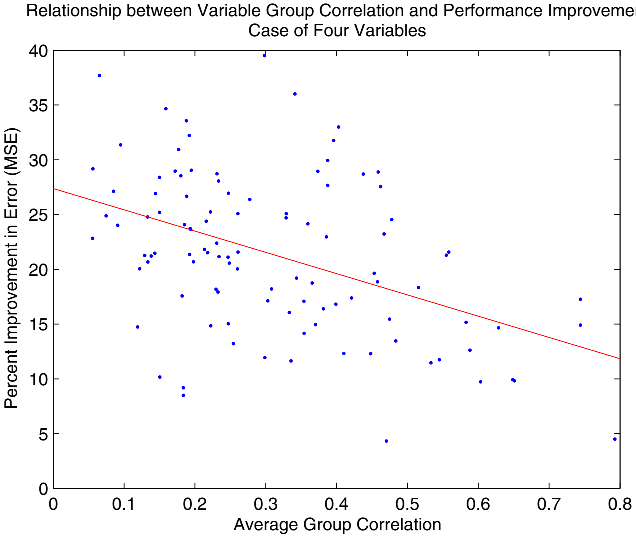


Fig. 6. The relationship between the group performance improvement and the average correlation coefficient among the variables in the selected group, for the case of selections of four variables. Each point represents a selection of variables selected at random from among the four benchmark problems considered. The performance improvement is measured in terms of the percent change in the mean square error. Also shown is the regression line for the points.

Figure 5 shows the case of selections of three input variables, while Figure 6 shows the case of selections of four input variables. The results obviously show the importance of correlation as a factor affecting group performance. This does not agree with the analysis performed last section for the linear regression case. For the neural network case, theroretically speaking, one can transform the variables linearly in a way to make the variables uncorrelated. This transformation matrix can be incorporated into first layer weights. Based on this argument, it should therefore be expected that correlations should *not* affect group performance. Since the simulation results clearly show the contrary, one might hypothesize that this is perhaps due to training having an easier time with less correlated input variables. Why this could be the case remains yet to be investigated.

8 Conclusions

In this research we have studied the role of individual input performance and correlations in predicting the performance of the selected group of inputs. First we considered the linear regression problem, and discovered through a simple example that individual performance is not a prerequisite for good group performance. However, good individual performance generally carries through to the group's performance. Also, we have proved that adding random or irrelevant inputs improves the performance in a linear fashion. We briefly also considered the correlation issue and proved that correlations do not tilt the expected performance one way or the other, even though high correlations can cause other problems in some other aspects. We have also observed through an empirical simulation study for multilayer networks that individual input performance does indeed affect group performance significantly. We have also shown that low correlation affects positively group performance, indicating the deviation from the linear regression case.

References

1. T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, John Wiley & Sons, Inc., 2003.
2. DELVE Data Repository, http://www.cs.toronto.edu/ delve/
3. M. Dong and R. Kothari, "Feature subset selection using a new definition of classifiability", *Pattern Recognition Letters*, Vol. 23, pp. 1215-1225, 2003.
4. F. Harrell, *Regression Modeling Strategies*, Springer-Verlag, 2001.
5. A. Jain , D. Zongker, "Feature Selection: Evaluation, Application, and Small Sample Performance", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 2, pp. 153-158, February 1997.
6. R. Kohavi , G. H. John, "Wrappers for feature subset selection", *Artificial Intelligence*, Vol. 97, No. 1-2, pp. 273-324, December 1997.
7. A. Naylor and G. Sell, *Linear Operator Theory in Science and Engineering*, Springer-Verlag, 1982.
8. L. Portinale and L. Saitta, "Feature selection", University of Dortmund Technical Report, Dortmund, Germany.
9. S. G. Yoo, K. T. Chong, and S. Y. Yi,"Neural netowrk modeling of transmission rate control factor for multimedia transmission using the Interent", APWeb2005, *Lecture Notes on Computer Science*, 3399, pp. 851-862, 2005.

Dimensionality Reduction of Protein Mass Spectrometry Data Using Random Projection

Chen Change Loy[1], Weng Kin Lai[1], and Chee Peng Lim[2]

[1] Grid Computing and Bioinformatics Lab, MIMOS Berhad, 57000 Kuala Lumpur, Malaysia
{chenchange.loy, lai}@mimos.my
[2] School of Electrical & Electronic Engineering, University of Science Malaysia, Engineering Campus, 14300 Nibong Tebal, Penang, Malaysia
cplim@eng.usm.my

Abstract. Protein mass spectrometry (MS) pattern recognition has recently emerged as a new method for cancer diagnosis. Unfortunately, classification performance may degrade owing to the enormously high dimensionality of the data. This paper investigates the use of Random Projection in protein MS data dimensionality reduction. The effectiveness of Random Projection (RP) is analyzed and compared against Principal Component Analysis (PCA) by using three classification algorithms, namely Support Vector Machine, Feed-forward Neural Networks and K-Nearest Neighbour. Three real-world cancer data sets are employed to evaluate the performances of RP and PCA. Through the investigations, RP method demonstrated better or at least comparable classification performance as PCA if the dimensionality of the projection matrix is sufficiently large. This paper also explores the use of RP as a pre-processing step prior to PCA. The results show that without sacrificing classification accuracy, performing RP prior to PCA significantly improves the computational time.

1 Introduction

For many types of cancer, the sooner the cancer is diagnosed and treated, the higher the survival rate is. Tumor markers such as cancer antigen 125 (CA125) and prostate-specific antigen (PSA) have been used widely as an early indicator of cancer. Tumor markers, however, may not have sufficient accuracy to reliably detect early stage cancer. A marker test that registers normal does not prove that a patient are cancer-free, nor does an elevated test prove that the patient have presence, progression or recurrence of cancer. Consequently, there is a critical need on new methods that are more reliable for early cancer detection [1].

Some researchers believe that cancer may affect the proteins or peptides concentration in human blood serum even in the early stages. The earliest attempt to prove this concept was carried out by Petricon et al. in 2002 [2]. They employed genetic algorithm combined with self-organizing maps to analyze the protein mass spectrometry (MS) pattern and was successful in discriminating ovarian cancer patients from unaffected individuals with an accuracy of 97.41%. Since then, the field of protein MS pattern recognition has been intensively researched, particularly focusing on early cancer detection.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 776 – 785, 2006.

The following is a typical work flow in protein MS pattern recognition process. Given an unknown sample of human blood serum, by using protein MS technology, a population of proteins in this sample is profiled based on the molecular mass-to-charge (*m*/*z*) identities of individual proteins. The output is a raw spectral that contains relative amplitudes of intensity at each *m*/*z* identity. By using pattern recognition techniques, researchers attempt to identify the pathological state of the unknown sample. Supervised pattern recognition techniques must learn from a set of training samples with known pathological states before it can generate prediction.

A critical challenge in MS pattern recognition is the extraction of concrete information from the MS data that can accurately reflect the pathological state. The difficulty lies in the fact that the MS data is usually characterized with small amount of samples and high-dimensional features. The typical ratio of samples to features is at the order of thousands. Learning in high dimensions causes problems that are either non-existent or less severe compared to lower-dimensional cases. Firstly, applying all the features introduces enormous computational overhead to the processing unit. Secondly, having relatively small amounts of training samples may lead to data overfitting, i.e. the predictor parameters are well optimized for the training samples but generalize poorly on new samples.

Generally, there are two approaches to overcome the problems, namely feature selection and dimensionality reduction. Feature selection is concerned with selecting a set of optimum discriminatory features that can reflect the actual biological classes represented in the data. In contrast to feature selection methods, dimensionality reduction techniques exploit the information from complete protein spectrum. The main idea of dimensionality reduction is to project the input onto a lower-dimensional space by preserving essential properties of the data. Common methods for dimensionality reduction include Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Partial Least Squares (PLS), Independent Component Analysis (ICA), and etc. Random Projection (RP) has lately emerged as an alternative method for dimensionality reduction. In fact, this technique has been tested on hand-written digit data set [3], image, and textual data [3] with fairly good results. In addition, RP was reported to be computationally less demanding compared with conventional dimensionality reduction techniques.

The objective of this study is to examine the effectiveness of RP in dimensionality reduction, particularly on protein MS data. Three real-world cancer data sets are used to achieve this. The performance of RP is compared with PCA using three classification methods, namely K-nearest Neighbour (KNN), Feed-forward Neural Networks (FFNN), and Support Vector Machine (SVM). Apart from that, experiments are also conducted to measure the distortion induced by RP and PCA. This study also investigates the use of RP as a pre-processing step prior to PCA.

The organization of this paper is as follows. A review of previous works is provided in Part 2. Part 3 gives an overview on PCA. The proposed RP method is then explained in detail. In Part 4, the data sets used in this paper are described. The results are reported and discussed in Part 5. Finally, the paper concludes with some suggestions for further investigation in Part 6.

2 Previous Works

Several dimensionality reduction strategies and classification methods have been proposed to analyze protein MS pattern from human blood serum. This section highlights some previous works that are using dimensionality reduction techniques on protein MS pattern. Other methods such as feature selection and peak detection are not covered here; interested readers can refer to [5] for further details.

In 2002, Lilien and co-workers developed a supervised classification method called Q5 for protein MS pattern recognition [6]. Q5 employed PCA and linear discriminant analysis followed by probabilistic classification. Q5 was tested against three ovarian cancer data sets and one prostate cancer data set. Replicate experiments of different training/testing partitions were carried out. The authors claimed that their algorithm achieved sensitivity, specificity, and accuracy above 97%.

In 2003, Purohit and Rocke conducted a comparative study of unsupervised method and supervised method on protein MS pattern of 41 patients [7]. Prior to the experiments, the authors progressively binned the data by averaging adjacent features within a uniform moving window. Square root transformation was then applied on the data. In the study of unsupervised classification, PCA combined with hierarchical clustering gave classification accuracy of 68%. Whereas in the study of supervised classification, PLS was used for dimensionality reduction. Two classification methods, namely linear discriminant analysis and logistic regression were proposed. Encouraging results were obtained and both of the suggested classification methods were claimed to have classification accuracy of 100% respectively. Leave-one-out cross validation was performed through out the experiments.

Similar experiments were undertaken by Shen and Tan in 2005 [8]. The authors also applied PLS for dimensionality reduction. Penalized logistic regression was proposed for classification. 30 random training/testing partitions were conducted to verify the classification results. By using the same ovarian cancer data set used in [6], the authors claimed that their approach have an accuracy of 99.92%. Sensitivity and specificity were not reported in their paper.

3 Dimensionality Reduction Techniques

3.1 Principal Component Analysis

One of the most widely used dimensionality reduction techniques is PCA. PCA aims at reducing the data dimensionality while determining orthogonal axes of maximal variance from the data. For PCA to work properly, the mean has to be subtracted from each dimension. Next, eigen decomposition of the covariance matrix is computed. Eigenvalues and eigenvectors are then sorted in descending order. Components with higher eigenvalues explain more of the total data variances. Normally, most of the variances are captured in the first few components. A new dimensionality-reduced data set can be derived by projecting the original data set onto these principal components. The projection matrix comprising these principal components is referred as “PCA basis” in this paper.

The main drawback of PCA is the computational complexity, which is known to be $O(d^2n)$ + $O(d^3)$, where d is data dimensionality and n is the number of cases. PCA is computationally costly because it performs the eigen decomposition of the covariance matrix. Although PCA may be carried out more efficiently by using SVD decomposition and by omitting zero eigenvalues in calculation, the computational overhead is still too high for high-dimensional data sets like protein MS data.

3.2 Random Projection

The main idea of RP originates from the Johnson-Lindenstrauss lemma. The theorem states that a set of n points in high-dimensional Euclidean space $\mathbb{R}^d$ can be projected onto a randomly chosen lower-dimensional Euclidean space $\mathbb{R}^k$ ($k < d$) without distorting the pairwise distances by more than a factor of ($1 \pm \varepsilon$). More precisely, according to the following Johnson-Lindenstrauss lemma [9]:

For any ε such that $0 < \varepsilon < ½$, and any set of points S in $\mathbb{R}^n$, with $|S| = m$, upon projection to a uniform random k-dimensional subspace, $k \geq [9 \ln m / (\varepsilon^2 - 2\varepsilon^3/3)] + 1$, the following property holds: with probability at least ½, for every pair $u, u' \in S$, and $f(u), f(u')$ are the projections of u, u'.

$$(1-\varepsilon)\left\|u-u'\right\|^2 \leq \left\|f(u)-f(u')\right\|^2 \leq (1+\varepsilon)\left\|u-u'\right\|^2 \tag{1}$$

The computational complexity of RP is lower than PCA, which may be expressed as $O(nkd)$. This is because of the steps to perform RP are mathematically simpler than PCA. To carry out the projection, a high-dimensional data matrix $\boldsymbol{X}_{d\times n}$ is multiplied with a projection matrix $\boldsymbol{R}_{k\times d}$. The projection matrix is a random orthogonal matrix where the Euclidean length of each column is normalized to unity. The resulting k-dimensional matrix $\boldsymbol{Y}_{k\times n}$ can be expressed as:

$$\boldsymbol{Y}_{k\times n} = \boldsymbol{R}_{k\times d}\boldsymbol{X}_{d\times n} \tag{2}$$

In most cases, the projection matrix is not completely orthogonal. However, by using a Gaussian distributed random matrix, whose entries is independent and identically distributed, with mean = 0 and variance = 1, the matrix would be very close to being orthogonal in a high-dimensional space. Therefore, a high-dimensional Gaussian distributed random matrix can be viewed as an approximation to an orthogonal matrix, in which the property can be expressed in the following equation:

$$\boldsymbol{R}^T\boldsymbol{R} \approx \boldsymbol{I} \tag{3}$$

In fact, there are simpler random distributions that have similar properties and yet computationally more efficient, such as sparse random matrices introduced by Achlioptas [10]. The entries in a sparse random matrix are either uniformly chosen from {-1, 1}, or from {$\pm\sqrt{3}$, 0}, by selecting $\pm\sqrt{3}$ with probability 1/6 each and 0 with probability 2/3. This paper will focus on the use of Gaussian distributed random matrix.

4 Data Sets

Three real-world cancer data sets, i.e. two ovarian cancer data sets (OC-WCX2a and OC-WCX2b) and one prostate cancer data set (PC-H4) were used to investigate the applicability of RP in reducing dimensionality of protein MS data. These data sets are obtained from Clinical Proteomics Program Databank, National Cancer Institute [11]. The data sets were named by using the cancer type screened and the SELDI affinity chip technology, i.e. Weak Cation Exchange (WCX2) and Hydrophobic (H4). OC-WCX2a and PC-H4 were manually prepared; OC-WCX2b was prepared by robotic instrument [11]. The data was generated by using surface-enhanced laser desorption/ionization time of flight (SELDI-TOF) mass spectrometer [2]. Each of these data sets consists of samples from cancer patients and control patients. Each sample is composed of 15154 features, which are defined by the corresponding molecular mass-to-charge (*m*/*z*) identities. All features were baseline subtracted and were rescaled so that they fall within the range of 0 and 1. The details of the data sets are summarized in Table 1.

Table 1. Details of cancer data sets

Data Set	Control	Cancer	Number of Features
OC-WCX2a	100	100	15154
OC-WCX2b	91	162	15154
PC-H4	63	69	15154

5 Results and Discussion

5.1 Distortion Analysis

Prior to the classification experiments, it is important to compare the distortion introduced by PCA and RP to the original data space. In order to compute the distortion, Equation (1) was rearranged, and the distortion $\boldsymbol{dist}_f(u, u')$ may be expressed as:

$$\boldsymbol{dist}_f(u,u') = \frac{\left\| f(u) - f(u') \right\|^2}{\left\| u - u' \right\|^2} \tag{4}$$

Distortion induced by PCA and RP is depicted in Fig. 1. The results were averaged over 100 pairs of random samples of OC-WCX2a data set. Unity $\boldsymbol{dist}_f(u, u')$ implies that there is no distortion induced. The lower the $\boldsymbol{dist}_f(u, u')$ is, the greater the distortion induced. As can be seen from Fig. 1, there is no significant difference between the distortion induced by PCA and RP within projection dimensionality from 10 to 140. This suggests that RP may perform as well as PCA by retaining a significant degree of data information.

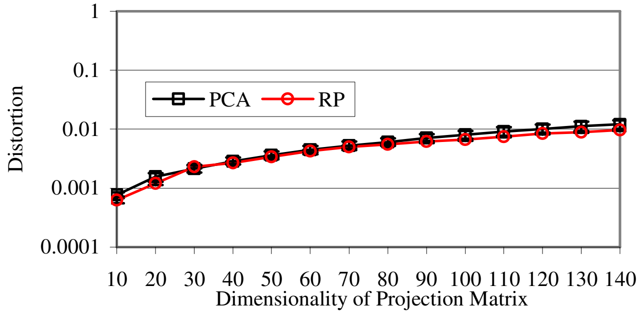


Fig. 1. This figure shows the results averaged over 100 pairs of random samples along with 95% confidence intervals

5.2 Classification Performance

A series of experiments have been conducted to compare the performance of RP with PCA by using three classification methods, namely KNN, FFNN, and SVM. The three learning algorithms were chosen so that they represent a diverse set of learning biases. The primary focus is not to compare the performance of these classification methods, but the differences in their performance while using PCA and RP for dimensionality reduction. Common performance metrics for medical diagnosis are used here, namely accuracy, sensitivity, and specificity. Positive case is referred to the presence of particular disease while a negative case means the absence of the disease. These metrics are calculated as follows:

- Accuracy – the ratio of the number of correctly diagnosed cases to the total number of cases.
- Sensitivity – the ratio of the number of positive cases correctly diagnosed to the total number of positive cases.
- Specificity – the ratio of the number of positive cases correctly diagnosed to the total number of positive cases.

Split-sample cross-validation was employed to validate the results. Using this validation method, a data set was randomly divided into training set and testing set. In this study, the samples for each data set were randomly partitioned into 75%/25% of training/testing set. The same process mentioned above was repeated to generate 30 random training/test partitions.

Following these partitions, dimensionality reduction was performed on each training set. For RP method, the original high-dimensional training samples were projected onto a lower-dimensional space by using a random matrix ***R***. Consequently, the dimensionality-reduced training samples were used to train the classifiers. In the testing stage, the same random matrix ***R*** was again used to project the original testing samples onto a lower-dimensional space. Similar procedures were repeated in the PCA experiments. After partitioning the training/testing samples, a PCA basis was computed from the training samples. The training samples were then projected onto the PCA-space representation via the PCA basis. In the testing stage, the testing samples were projected onto the PCA-space representation by using the same PCA basis. Each classifier's performance was measured based on the predictions of the dimensionality-reduced testing samples. Note that all testing sets were not involved in the prediction model building process in order to avoid biased results.

For both RP and PCA experiments, the dimensionality of projection matrix, i.e. random matrix ***R*** and PCA basis was changed from low value (10) to a high value (140) in order to examine the effect of this parameter on the results. For each of these different dimensionalities, 30 random training/test partitions were tested and the final results were estimated using 1000 bootstrap samples at 95% confidence intervals. The results are given in Fig. 2. Sensitivity, specificity, and confidence intervals are not shown in Fig. 2 so as to maintain clarity and readability. However, exceptional cases will be reported and discussed in the paper.

Prior to the KNN experiment, some experiments were carried out to find the optimum number of neighbours *K*. It turned out that $K = 5$ gave the best performance, so this value was used here. As can be observed from Fig. 2, the performance of PCA generally outperformed RP in the lower dimensions. Nevertheless, performance of PCA showed a drastic decline when the dimensionality increased. The performance of KNN appeared to be less affected by RP. The results may be expected since the underlying operation of KNN is based on Euclidean distance computations, while RP tends to preserve the inter-point distances. Thus, one would expect to obtain better results by using RP as compared with PCA.

For the FFNN experiment, the number of hidden units was fixed to five. In contrast to KNN, performance of PCA remained constant throughout the experiment. RP performed slightly poorer as compared with PCA, but its performance improved noticeably as the dimensionality of projections increased. Performance of PCA was better in lower dimensions, but RP was able to match the performance of PCA in higher dimensions.

Apart from KNN and FFNN, SVM with Gaussian radial basis kernel was also employed in this study. From Fig. 2, it can be observed that performance of PCA was generally better than RP in lower dimensions, but the performance of RP was better than PCA or at least comparable with PCA when the dimensionality increased. RP yielded the best results when it was combined with SVM, and the dimensionality of projection matrix was set to 100. The details of the results are summarized in Table 2 along with the 95% confidence intervals in parentheses.

Table 2. Classification performance of RP combined with SVM

Data Set	Accuracy (%)	Sensitivity (%)	Specificity (%)
OC-WCX2a	94.00 (94.00 - 94.00)	95.87 (95.60 - 96.00)	92.13 (92.00 - 92.40)
OC-WCX2b	99.89 (99.68 - 100)	100.00 (100 - 100)	99.71 (99.13 - 100)
PC-H4	99.90 (99.70 - 100)	99.81 (99.41 - 100)	100.00 (100 - 100)

On the whole, the performance of PCA remained approximately stable and outperformed RP results in the lower dimensions, but its performance decreases in higher dimensions. Although the performance of RP was inferior to the performance of PCA in lower dimensions, the performance improved as dimensionality increased. The drop of PCA average accuracies in higher dimensions may be caused by the inclusion of less important components that leads to worse rather than better performance.

5.3 Combination of PCA and RP

Due to the ability of keeping the subspace that has largest variance, PCA seemed to be better in eliminating the impact of noise in a data set to some extent. This advantage, however, comes at the price of greater computational requirement, especially for high-dimensional data sets. On the other hand, RP is computationally more efficient for high-dimensional data set but it may not be able to filter out redundant information. The main purpose of this experiment is to investigate the performance of combining PCA and RP with the intention to complement the strength of both techniques. In this experiment, RP was performed prior to PCA to reduce the original data dimensionality from 15154 to 140. Then, PCA was carried out in the lower-dimensional space to eliminate the redundant information.

As can be seen from Fig. 2, in most cases, accuracies of PCA + RP were comparable to PCA results and RP results in low dimensions. Table 3 summarizes the average processing time taken for PCA, RP and their combination. The amount of time to perform RP and PCA+RP is clearly shorter than using PCA alone. RP considerably speeds up PCA algorithms whose run-time is largely governed by the dimension of the working space. The results suggested that RP could be used as pre-processing step before PCA in order to reduce the computational load without introducing great distortions to the original data.

Table 3. Average Processing Time Over 30 Runs Using OC-WCX2a Data Set

Method	Processing Time (sec)
PCA	16.571 ± 0.213
RP	3.945 ± 0.044
PCA + RP	4.016 ± 0.048

6 Conclusions and Further Works

Protein MS technology allows medical practitioners to characterize and determine the patterns of tens of thousands of proteins simultaneously. Unfortunately, conventional dimensionality reduction methods and pattern recognition techniques may fail because of the high dimensionality of the data. The work presented in this paper investigates the efficacy of using RP as a dimensionality reduction tool for protein MS data. A series of experiments have been systematically conducted to compare the performance of RP with PCA. Performances of PCA and RP were tested against three cancer data sets by using KNN, FFNN, and SVM. From the results, performance of RP generally improves as the dimensionality increases. Although PCA slightly outperformed RP in low dimensions, RP was able to achieve comparable performance in high-dimensional space, and yet with less computational overhead.

Another focus of this paper is to explore the use of RP as a pre-processing step prior to PCA. As a result of performing RP beforehand, PCA took advantage of the speedup produced by working over fewer dimensions. In most cases, the accuracies obtained were comparable to PCA results and RP results in low dimensions.

The work presented in this paper has revealed the potential of RP as an efficient dimensionality reduction method for protein MS data. Nonetheless, there are still a number of areas that can be enhanced and pursued as further work. Firstly, experiments can be carried out to investigate the use of sparse random matrices [10]. Apart from that, another issue that needs to be addressed is the drifts in SELDI-TOF machines [15]. Protein MS data generated by SELDI-TOF machines may vary time-to-time and machine-to-machine. Current experiments are based on low-resolution MS data. Motivated by the need for greater precision, more experiments are undergoing to further vindicate the proposed RP method by using high-resolution MS data [15]. But then again, higher data resolution propagates the "curse of dimensionality" and increases the computational overhead. Intuitively, one would expect the computational requirement of PCA to increase exponentially. Again, RP may play an important role in reducing the dimensionality of high-resolution MS data.

Protein MS pattern analysis has great potential for use as part of standard cancer diagnosis tests. Nevertheless, in terms of practicality, there are a number of challenges that have to be resolved before MS pattern recognition can be fully applied in medical screening test or treatment monitoring. The results shown in this paper revealed that RP may be an alternative to conventional dimensionality reduction techniques for high-dimensional protein MS data. Nonetheless, RP is not restricted for protein MS data, but also other high-dimensional data such as textual data and microarray data.

References

1. Perkins, G.L., et al.: Serum Tumor Markers. American Family Physician, Vol. 68, No. 6. (2003) 1075–1082
2. Petricon, E.F., et al.: Use of Proteomic Patterns in Serum to Identify Ovarian Cancer. The Lancet, Vol. 359. (2002) 572–577
3. Dasgupta, S.: Experiments with Random Projections. Proc. 16th Conf. Uncertainty in Artificial Intelligence. (2000)
4. Bingham, E., Mannila, H.: Random Projection in Dimensionality Reduction Application to Image and Text Data. Knowledge Discovery and Data Mining. (2001) 245–250
5. Levner, I.: Feature Selection and Nearest Centroid Classification for Protein Mass Spectrometry. Bioinformatics, Vol. 6, No. 68. (2005)
6. Lilien, R.H., Farid, H., and Donald, B.R.: Probabilistic Disease Classification of Expression-Dependent Proteomic Data from Mass Spectrometry of Human Serum. J. of Computational Biology, Vol. 10, No. 6. (2003) 925–946
7. Shen, L., and Tan, E.C.: Dimension Reduction-Based Penalized Logistic Regression for Cancer Classification using Microarray Data. IEEE/ACM Trans. on Computational Biology and Bioinformatics, Vol. 2, No. 2. (2005) 166–174
8. Purohit, P.V., and Rocke, D.M.: Discriminant Models for High-Throughput Proteomics Mass Spectrometer Data. Proteomics, Vol. 3. (2003) 1699–1703
9. Vempala, S.S.: The Random Projection Method. Vol. 65. American Mathematical Society (2004)
10. Achlioptas, D.: Database-Friendly Random Projections. Symposium on Principles of Database Systems. (2001) 274–281

11. Clinical Proteomics Program Databank, National Cancer Institute: http://home.ccr.cancer.gov/ncifdaproteomics/ppatterns.asp.
12. Conrads, T.P., et al.: High-Resolution Serum Proteomic Features for Ovarian Cancer Detection. Endocrine-Related Cancer, Vol. 11. (2004) 163–178

Appendix. Classification Performance

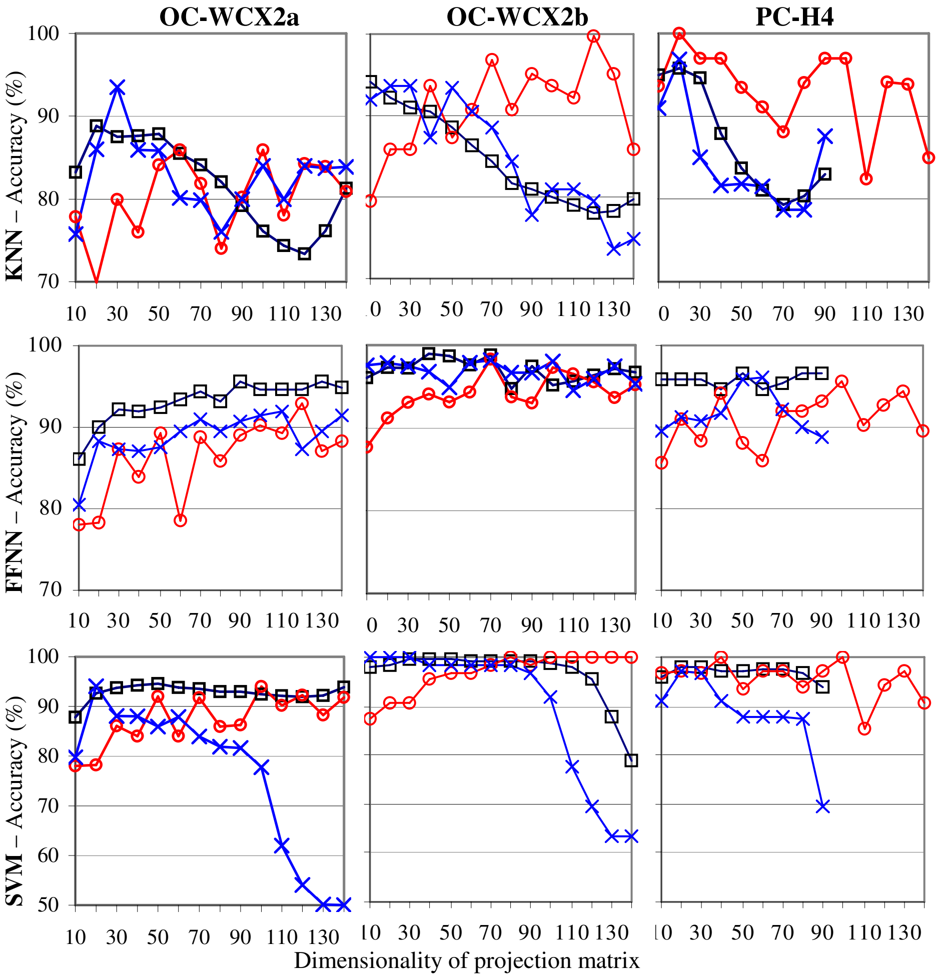


Fig. 2. This figure shows the accuracies (Y-axis) of KNN, FFNN, and SVM when three different dimensionality reduction techniques are used, namely PCA (□), RP (○), and PCA+RP (×). X-axis represents the dimensionality of projection matrix. Results are averaged over 30 training/testing partitions and are estimated using bootstrapping method at 95% confidence intervals.

Fault Tolerant Training of Neural Networks for Learning Vector Quantization

Takashi Minohara

Dept. of Computer Science, Takushoku University
815-1, Tatemachi, Hachioji, Tokyo, 193-0985, Japan
minohara@cs.takushoku-u.ac.jp

Abstract. The learning vector quantization(LVQ) is a model of neural networks, and it is used for complex pattern classifications in which typical feedforward networks don't give a good performance. Fault tolerance is an important feature in the neural networks, when they are used for critical application. Many methods for enhancing the fault tolerance of neural networks have been proposed, but most of them are for feedforward networks. There is scarcely any methods for fault tolerance of LVQ neural networks. In this paper, I proposed a dependability measure for the LVQ neural networks, and then I presented two idea, the border emphasis and the encouragement of coupling, to improve the learning algorithm for increasing dependability. The experiment result shows that the proposed algorithm trains networks so that they can achieve high dependability.

1 Introduction

Nowadays, artificial neural networks are studied and used in various applications, for example, in pattern and speech recognition, in classification, in signal and image processing and so on[1]. Advanced VLSI technologies allows the hardware implementation of such systems at reasonable costs and with respectable performance. Since defects at the end of production nor faults during life time are not avoidable in hardware, fault tolerance is an important feature when they are used in mission-critical applications.

It is often claimed that neural networks are inherently fault tolerant, because neural networks are distributed computing systems, and are insensitive to partial internal faults. However, without a special design, it is difficult to guarantee the degree of fault tolerance. Martin and Damper[2] shows that an increase in the number of nodes will not ensure improvement in fault tolerance. Some mechanisms to enhance the fault tolerance should be incorporated into the implementation.

Neural networks are categorized into the three models: signal-transfer networks, state-transfer networks and competitive-learning networks. In the signal-transfer networks, the output signal values depend uniquely on input signal. Typical representatives of this model are layered feed forward networks in which the error-back-propagation algorithm is used for learning. In the state-transfer

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 786–795, 2006.

networks, the feedbacks from the output signal values change the state of networks, and it converges to one of its stable states. Typical representatives of this model are the Hopfield networks[3] and the Boltzmann machine[4]. In the competitive-learning networks, the neurons receive identical input information, and they compete in their activities. Typical representatives of this model are the self-organizing maps(SOM)[5] and the learning vector quantization(LVQ)[5]. The SOM and LVQ are used for complex pattern classifications[6] in which feed forward networks don't give a good performance because of the difficulty of convergence with error-back-propagation. The schemes to increase fault-tolerance of neural networks have been proposed in literature[7][8][9], but most of them are focused only on the feedforward networks, i.e. fault tolerance in competitive-learning networks is out of work.

In this paper, after a brief review of the LVQ, I discuss the effects of faults and dependability measure in the LVQ. In section 4, I propose a training algorithm for improving the fault-tolerant capabilities of the LVQ. In section 5, I present experimental results of proposed methodology.

2 Brief Review of Learning Vector Quantization

Learning Vector Quantization(LVQ) is a method of a statistical classification or recognition, and its purpose is to define class region in the input space $\Re^n$. Figure 1 shows a configuration of the LVQ neural network system. Each neuron N_i has a *codebook vector* $\mathbf{m}_i = [\mu_{i1}, \mu_{i2}, \cdots, \mu_{in}]^T \in \Re^n$, and is assigned to one of the classes into which *input vector* $\mathbf{x} = [x_1, x_2, \cdots, x_n]^T \in \Re^n$ is classified. An input vector $\mathbf{x}$ is given to all neurons in parallel, and each neuron calculates the distances between $\mathbf{x}$ and its codebook vectors $\mathbf{m}_i$. Thereafter, $\mathbf{x}$ is determined to belong to the same class to which the nearest $\mathbf{m}_i$ belongs.

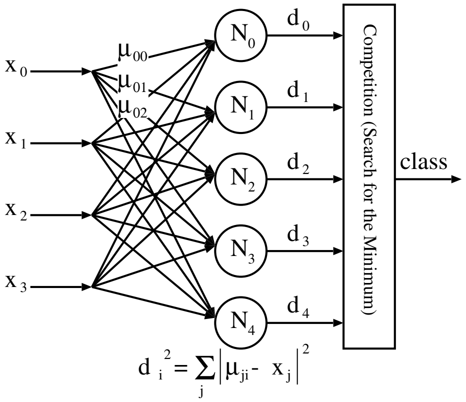


Fig. 1. Learning Vector Quantization neural networks

The value of $\mathbf{m}_i$ is obtained by the following learning process. Let c be the index of the nearest $\mathbf{m}_i$ to $\mathbf{x}$:

$$c = arg \min_i \{d(\mathbf{x}, \mathbf{m}_i)\} \tag{1}$$

where $d(\mathbf{x}, \mathbf{m}_i)$ is a Euclidean distance $||\mathbf{x} - \mathbf{m}_i||$. And let $x(t)$ be an input sample, and let $m_i(t)$ represent sequential values of the m_i in the discrete-time domain, $t = 0, 1, 2, \cdots$. The following equations define the LVQ process.

$$\mathbf{m}_c(t+1) = \mathbf{m}_c(t) + \alpha_c(t)[\mathbf{x}(t) - \mathbf{m}_c(t)]$$
$$\text{if } \mathbf{x} \text{ and } \mathbf{m}_\mathrm{c} \text{ belong to the same class} \tag{2}$$
$$\mathbf{m}_c(t+1) = \mathbf{m}_c(t) - \alpha_c(t)[\mathbf{x}(t) - \mathbf{m}_c(t)]$$
$$\text{if } \mathbf{x} \text{ and } \mathbf{m}_\mathrm{c} \text{ belong to different classes} \tag{3}$$
$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t)$$
$$\text{for } i \neq c \tag{4}$$

Here $0 < \alpha_i(t) < 1$, and $\alpha_i(t)$ (learning rate) is decreased monotonically with time (in LVQ1 algorithm) or optimized by the following recursion (in OLVQ1 algorithm)

$$\alpha_c(t) = \frac{\alpha_c(t-1)}{1 + s(t)\alpha_c(t-1)} \tag{5}$$

where $s(t) = 1$ if the classification is correct, and $s(t) = -1$ otherwise.

In LVQ1 and OLVQ1 learning algorithm, only one codebook vector which is the closest to the input vector(i.e. the winner's codebook) is updated. The improved learning algorithms have been proposed in which not only the winner's codebook but also the runner-up's codebook is updated. Here we introduce one of them that is called LVQ3. Suppose $\mathbf{m}_i$ and $\mathbf{m}_j$ are the two closest codebook vectors to the input vector $\mathbf{x}$. They are updated as follows if $\mathbf{x}$ and $\mathbf{m}_i$ belong to the same class, while $\mathbf{x}$ and $\mathbf{m}_j$ belong to the different classes.

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{m}_i(t)]$$
$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) - \alpha(t)[\mathbf{x}(t) - \mathbf{m}_j(t)] \tag{6}$$

If all of three vector $\mathbf{x}$, $\mathbf{m}_i$ and $\mathbf{m}_i$ belong to the same class, both codebook vector are updated toward the input vector $\mathbf{x}$ with learning factor ϵ.

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \epsilon\alpha(t)[\mathbf{x}(t) - \mathbf{m}_i(t)]$$
$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) + \epsilon\alpha(t)[\mathbf{x}(t) - \mathbf{m}_j(t)] \tag{7}$$

3 Fault Model and Effects of the Faults in LVQ

Due to a wide variety of hardware implementation of neural networks, it is difficult to find a general representation of faults in the lower level of system architecture. In this paper, I consider the behavior of neural network system in the functional level, and assume the following fault model.

- I assume fault occur on neurons. The faulty neuron loses its functionality and never be selected as the nearest codebook vector.
- I assume no fault occurs on competition mechanism. The input vector is classified into the class to which the nearest fault-free neuron(codebook vector) belongs.

Because of the competitive nature of LVQ neural networks, only the neuron which is the nearest to the input vector concerns output. Therefore, it depends on the input vectors whether a fault causes erroneous classification or not. I need a fault tolerance metric related to input vectors.

Under the fault models above, a fault is activated only when the faulty neuron is the nearest to the input vector, and belongs to the class different from the runner-up neuron. Let $P(F_i)$ be the a priori probability of fault F_i, where F_i denote the fault on the neuron N_i, and let $p(\mathbf{x})$ be the probability density function of the input vector $\mathbf{x}$. Then, the expected error rate E is given by

$$E = \sum_i \int s(F_i, \mathbf{x}) P(F_i) p(\mathbf{x}) d\mathbf{x} \tag{8}$$

where $s(F_i, \mathbf{x})$ is the activity of the fault by the input vector;

$$\begin{aligned} s(F_i, \mathbf{x}) = 1 \quad & \text{if } \mathrm{N_i} \text{ is the nearest to } \mathbf{x}, \text{and} \\ & \mathrm{class(N_i)} \neq \text{class(the runner-up neuron)} \quad (9) \\ s(F_i, \mathbf{x}) = 0 \quad & \text{otherwise} \quad (10) \end{aligned}$$

Unfortunately, in most application, It is rarely obtained that the knowledge about the probability density function of the input vector $p(\mathbf{x})$. In a typical case there is only a number of samples or training data. Supposing that sample data set X consists of m samples $\mathbf{x}_1, \cdots, \mathbf{x}_m$, and they are drawn independently and identically distributed according to the probability law $p(\mathbf{x})$, I introduce $E(X)$, the expected error for sample data set X, as follows.

$$E(X) = \sum_i \sum_{j=1}^{m} s(F_i, \mathbf{x_j}) P(F_i) \tag{11}$$

Assume that the probabilities of fault $P(F_i)$ are identical for all neurons, then $E(X)$ is given by;

$$E(X)|_{P(F_i)=\lambda} = \lambda \sum_i \sum_{j=1}^{m} s(F_i, \mathbf{x_j}) \tag{12}$$

I define the coefficient of error for sample data set X as;

$$CE(X) = \sum_i \sum_{j=1}^{m} s(F_i, \mathbf{x_j}) \tag{13}$$

In order to evaluate the fault tolerance of LVQ neural networks, I have calculated $CE(X)$ for the networks trained by OLVQ1 algorithm in the programming

Table 1. Fault tolerance of the networks trained by the Iris plants database

# of Neurons		8	9	10	11	12	13
Running Length of Training		400	600	800	1,000	1,200	1,400
Total Accuracy	(mean)	96.73	96.73	97.13	96.80	97.47	97.33
w/o Faults(%)	(s.d.)	0.36	0.38	0.35	0.42	0.24	0.23
$CE(X)$	(mean)	25.5	19.9	19.5	20.2	14.2	17.4
	(s.d)	1.78	1.79	1.16	1.88	1.49	1.72

Table 2. Fault tolerance of the networks trained by the speech signal database

# of Neurons		150	200	250	300	350	400
Running Length of Training		6,000	8,000	10,000	12,000	14,000	16,000
Total Accuracy	(mean)	92.88	93.73	94.32	94.81	95.10	95.31
w/o Faults(%)	(s.d.)	0.10	0.14	0.08	0.10	0.03	0.05
$CE(X)$	(mean)	201.1	183.9	181.0	175.9	168.6	162.3
	(s.d.)	4.92	2.34	2.85	4.80	2.12	3.44

package "LVQ_PAK"[5][10]. The training data sets I used are the iris plants database[11] and the speech signal database[10]. The former has 150 instances with 4 attributes and categorized in 3 classes, and the latter has 1,962 instances with 20 attributes and categorized in 20 classes. Table 1 and 2 show the results of calculation. As shown in these tables, the CE(X)s of the networks doesn't decrease very much even if the number of neurons increased, and the redundant neurons seem not to contribute to the improvement of fault tolerance.

4 Fault Tolerant Training of LVQ

With my fault models, the fault causes a failure only when the faulty neuron is the nearest to the input vector, and it belongs to the class different from the runner-up neuron. Thus the influence of the faults around the class border is more serious than the one in the center of the class. On the other hand, the influence of the faults becomes smaller as the density of the codebook vectors which belongs to same class is higher, because the function of the faulty neuron may be covered by the neuron of the same class. In extreme case, A single fault is masked by the duplication of neurons.

4.1 Border Emphasis

As mentioned above, form the point of view of fault tolerance, the density of the codebook vector should be high around the class border. However, the LVQ learns a codebook vector so that it approximates the probability density function of the input data space. If the probability density of the target problem is high in the center of the decision area, and low around the borders, the fault tolerance of LVQ neural networks couldn't be expected.

The purpose of LVQ is to define class regions in the input data space, only the codebook vectors that is closest to the class borders are important to the decision, a good approximation of the probability density function is not necessary. I propose the "border emphasis" learning to increase the fault tolerance of LVQ neural networks. I change the distribution of the original input data, so that the network learns the input vectors which are close to the borders repeatedly, and the density of the codebook vector becomes higher around the borders.

Unfortunately, I can't know which input vectors are close to borders, in advance. Instead, I assume that an input vector is near to the borders if the class of the winner's codebook vector is different from the class of the runner-up vector. Vectors which meet my assumption propose are marked, and the network learns the marked vectors again.

4.2 Encouragement of Coupling

The single fault in the neurons will be masked by the duplication, and the duplicated network can be composed as follows;

1. Make each two neuron a pair.
2. Give the same initial value to each of the neuron pair.
3. Update simultaneously each of the neuron pair.

However the duplication requires twice as much hardware as original one without improving the accuracy of classification. It is not necessary to duplicate all neurons because there exists the neurons which aren't sensitive to the faults.

In the LVQ3 algorithm, as shown in equation (7), the both of the winner and the runner-up update their codebook vectors towards the input vector when they belong to the same class. It has the effect which collects two codebook vectors in one place. I noticed this effect and propose the improvement of the LVQ3 algorithm which encourage the coupling of two codebook vectors.

I encourage the coupling by enlarging the ϵ in (7) when the input vector is close to the border of the classes. Again, I have no knowledge about the position of the border before learning. I suppose the input vector is close to the border, if the third place of the codebook vector belongs to the class different from the winner and the runner-up.

5 Experimental Result

I implement the proposed learning algorithm as the improvement of the LVQ3 algorithm. I call them FTLVQ3.1, FTLVQ3.2 and FTLVQ3.3 respectively as follows:

FTLVQ3.1 LVQ3 + Border emphasis
FTLVQ3.2 LVQ3 + Encourage of coupling
FTLVQ3.3 LVQ3 + Border emphasis + Encourage of coupling

As an example, The detail of FTLVQ3.3 is shown in the appendix.

I evaluate these algorithms on classification problems. The training sets are the same as what are used in section 3. Every neuron is labeled so that each class has the same number of codebook vectors, and the codebook vectors are initialized with the training data set. As training parameter, I setup initial value learning-rate $\alpha = 0.05$, and $\epsilon = 0.1$. In FTLVQ3.2 and FTLVQ3.3, ϵ is enlarged to 1.0 when the coupling is encouraged.

Table 3. Coefficient of error and accuracy(%) of the classification for the iris plants database

# of Neurons			8	9	10	11	12	13
OLVQ1	CE(X)	(mean)	25.5	19.9	19.5	20.2	14.2	17.4
		(s.d)	1.78	1.79	1.16	1.88	1.49	1.72
	accuracy	(mean)	96.73	96.73	97.13	96.80	97.47	97.33
		(s.d.)	0.36	0.38	0.35	0.43	0.25	0.23
LVQ3	CE(X)	(mean)	13.2	12.3	9.3	8.5	3.4	4.8
		(s.d)	0.83	0.55	1.07	1.12	0.53	0.80
	accuracy	(mean)	97.67	97.73	97.47	97.80	97.60	97.80
		(s.d.)	0.11	0.14	0.13	0.16	0.10	0.17
FTLVQ3.1	CE(X)	(mean)	9.2	9.3	8.3	6.9	4.6	6.7
		(s.d)	0.61	0.40	0.58	0.73	0.38	0.93
	accuracy	(mean)	97.53	98.00	98.00	98.00	98.07	98.20
		(s.d.)	0.19	0.00	0.00	0.19	0.15	0.10
FTLVQ3.2	CE(X)	(mean)	1.6	2.6	1.9	0.9	0.8	0.6
		(s.d)	0.32	0.45	0.50	0.30	0.19	0.25
	accuracy	(mean)	94.60	97.67	97.60	97.07	97.93	97.53
		(s.d.)	0.93	0.11	0.34	0.30	0.06	0.10
FTLVQ3.3	CE(X)	(mean)	4.1	3.3	3.5	2.9	1.0	1.1
		(s.d)	0.52	0.60	0.75	1.17	0.64	0.50
	accuracy	(mean)	97.20	97.93	98.00	97.47	97.87	98.00
		(s.d.)	0.53	0.06	0.94	0.32	0.25	0.13

Table 3 and Table 4 shows the results of my experiments on the accuracy of the classification and the calculated value of $CE(X)$ for the training data sets. For both data sets, the encourage of coupling(FTLVQ3.2) shows good performance with relatively small number of neurons are employed. Although the Border emphasis(FTLVQ3.1) is not good for the iris plants database, It has got the best performance for the speech signal database with relatively large number of neurons. The both methods works complementarily in the FTLVQ3.3, improvements of $CE(X)$ are observed for every case of my experiments.

Table 4. Coefficient of error and accuracy of the classification for the speech signal database

# of Neurons			150	200	250	300	350	400
OLVQ1	CE(X)	(mean)	201.1	183.9	181.0	175.9	168.6	162.3
		(s.d.)	4.92	2.34	2.85	4.80	2.12	3.44
	accuracy	(mean)	92.88	93.73	94.32	94.81	95.10	95.10
		(s.d.)	0.10	0.14	0.08	0.10	0.03	0.05
LVQ3	CE(X)	(mean)	88.0	61.5	47.9	42.4	32.7	31.0
		(s.d.)	4.28	2.41	2.71	2.40	0.65	1.85
	accuracy	(mean)	94.63	94.99	95.14	95.30	95.30	95.39
		(s.d.)	0.12	0.11	0.10	0.10	0.09	0.08
FTLVQ3.1	CE(X)	(mean)	107.9	61.0	40.4	26.3	16.8	11.3
		(s.d.)	5.19	3.53	4.04	2.92	1.46	0.92
	accuracy	(mean)	95.85	96.12	96.12	96.06	96.04	96.02
		(s.d.)	0.06	0.06	0.12	0.10	0.11	0.08
FTLVQ3.2	CE(X)	(mean)	61.6	49.2	44.9	46.1	39.5	36.9
		(s.d)	2.02	2.93	1.66	2.61	2.66	1.93
	accuracy	(mean)	93.92	94.62	95.06	95.51	95.57	95.78
		(s.d.)	0.14	0.08	0.05	0.08	0.09	0.08
FTLVQ3.3	CE(X)	(mean)	63.2	45.2	30.9	26.1	17.1	13.9
		(s.d)	4.87	5.09	2.72	2.68	2.63	1.07
	accuracy	(mean)	95.47	96.02	96.22	96.57	96.41	96.42
		(s.d.)	0.15	0.16	0.14	0.10	0.11	0.05

6 Conclusion

In this paper I proposed the coefficient of error($CE(X)$) as a dependability measure for the LVQ neural networks, and also presented two idea, the border emphasis and the encouragement of coupling, to improve the learning algorithm for increasing dependability. The experiment result shows that the former idea is effective with large number of neurons, while the latter one shows good performance with small number of neurons, and they work complementarily for training networks so that they can achieve high dependability. I excluded the faults in the competition mechanism from the fault model of this paper, it remains as a future subject.

References

1. Herts, J., Krogh, A., Palmer, R.G.: Introduction to the Theory of Neural Computation. Addison-Wesley (1991)
2. Martin, D.E., Damper, R.I.: Determining and improving the fault tolerance of multilayer perceptrons in a pattern-recognition application. IEEE Trans. on Neural Networks **4**(5) (1993) 788–793
3. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA **79** (1982) 2254–2558

4. Fahlman, S.E., Hinton, G.E.: Massively parallel architectures for AI: NETL, Thistle, and Boltzmann Machines. In: Proceedings of the National Conference on Artificial Intelligence AAAI-83. (1983) 109–113
5. Kohonen, T.: Self-Organizing Maps. Springer-Verlag (1995)
6. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. second edn. John Wiley & Sons, Inc (2001)
7. Tan, Y., Nanya, T.: Fault-tolerant back-propagation model and its generalization ability. In: Digest IJCNN. (1991) 1373–1378
8. Phatak, D.S.: Fault tolerant artificial neural networks. In: 5th Dual Use Technologies and Applications Conference. (1995) 193–198
9. Hammadi, N.C., Ohmameuda, T., Kaneko, K., Ito, H.: Fault tolerant constructive algorithm for feedforward neural networks. In: PRFTS'97. (1997) 215–220
10. LVQ Programming Team of the Helsinki University of Technology: LVQ_PAK: The learning vector quantization program package. `ftp://cochlea.hut.fi/pub/lvq_pak/` (1995)
11. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annual Eugenics **Part II**(7) (1936) 179–188

Appendix. FTLVQ3.3 Algorithm

```
$mode \Leftarrow$ "$normal$" ; $j \Leftarrow 1$ ; $k \Leftarrow 0$ ; $l \Leftarrow 1$
while ( $l \leq l_{runlength}$ )
loop: /* The selection of the input vector */
    if ( mode = "normal")
        if ($j <= N_{sampledata}$ )
            $\mathbf{x} = \mathbf{x}_j$
        else
            $mode \Leftarrow$ "$additional$"
            goto loop
    else
        if ($k >= 1$ )
            $\mathbf{x} = \mathbf{y}_k$
        else
            $mode \Leftarrow$ "$normal$"
            $j = 1$
            goto loop
/* Calculate the distance between codebook vectors and the input vector */
    for ($i \Leftarrow 1$; $j \leq N_{neuron}$; $i \Leftarrow i + 1$)
        $d_i \Leftarrow ||\mathbf{m}_i - \mathbf{x}||$
/* Search for codebook vectors which are the closest to the input,
the runner-up, and the third place. */
    if ( $d_1 < d_2$ )
        if ( $d_2 < d_3$ )
            $c \Leftarrow 1$; $r \Leftarrow 2$ ; $t \Leftarrow 3$
        else if ( $d_1 < d_3$ )
            $c \Leftarrow 1$; $r \Leftarrow 3$ ; $t \Leftarrow 2$
        else
```

```
            $c \Leftarrow 3$; $r \Leftarrow 1$ ; $t \Leftarrow 2$
    else if ( $d_1 < d_3$ )
        $c \Leftarrow 2$; $r \Leftarrow 1$ ; $t \Leftarrow 3$
    else if ( $d_2 < d_3$ )
        $c \Leftarrow 2$; $r \Leftarrow 3$ ; $t \Leftarrow 1$
    else
        $c \Leftarrow 3$; $r \Leftarrow 2$ ; $t \Leftarrow 1$
    for ($i \Leftarrow 4$; $i \leq N_{neuron}$; $i \Leftarrow i + 1$)
        if ( $d_i < d_c$ )
            $t \Leftarrow r$ ; $r \Leftarrow c$ ; $c \Leftarrow i$
        else if ( $d_i < d_r$ )
            $t \Leftarrow r$ ; $r \Leftarrow i$
        else if ( $d_i < d_t$ )
            $t \Leftarrow i$
/* The update of the codebook vectors */
    if ( class($\mathbf{m}_c$) $\neq$ class($\mathbf{m}_r$) )
        if ( mode = "normal")
            $\mathbf{y}_k \Leftarrow \mathbf{x}$ ; $k \Leftarrow k + 1$
        if ( class($\mathbf{m}_c$) = class($\mathbf{x}$) )
            $\mathbf{m}_c \Leftarrow \mathbf{m}_c + \alpha(l)(\mathbf{x} - \mathbf{m}_c)$
            $\mathbf{m}_r \Leftarrow \mathbf{m}_r - \alpha(l)(\mathbf{x} - \mathbf{m}_r)$
        if ( class($\mathbf{m}_r$) = class($\mathbf{x}$) )
            $\mathbf{m}_c \Leftarrow \mathbf{m}_c + \alpha(l)(\mathbf{x} - \mathbf{m}_c)$
            $\mathbf{m}_r \Leftarrow \mathbf{m}_r - \alpha(l)(\mathbf{x} - \mathbf{m}_r)$
    else if ( class($\mathbf{m}_c$) = class($\mathbf{x}$) )
        if ( class($\mathbf{m}_t$) $\neq$ class($\mathbf{x}$) )
            $\mathbf{m}_c \Leftarrow \mathbf{m}_c + \alpha(l)(\mathbf{x} - \mathbf{m}_c)$
            $\mathbf{m}_r \Leftarrow \mathbf{m}_r + \alpha(l)(\mathbf{x} - \mathbf{m}_r)$
        else
            $\mathbf{m}_c \Leftarrow \mathbf{m}_c + \epsilon \cdot \alpha(l)(\mathbf{x} - \mathbf{m}_c)$
            $\mathbf{m}_r \Leftarrow \mathbf{m}_r + \epsilon \cdot \alpha(l)(\mathbf{x} - \mathbf{m}_r)$
    $l \Leftarrow l + 1$
    if ( mode = "normal")
        $j \Leftarrow j + 1$
    else
        $k \Leftarrow k - 1$
```

Clustering with a Semantic Criterion Based on Dimensionality Analysis

Wenye Li, Kin-Hong Lee, and Kwong-Sak Leung

The Chinese University of Hong Kong,
Shatin N.T., Hong Kong, China P.R.
{wyli, khlee, ksleung}@cse.cuhk.edu.hk

Abstract. Considering data processing problems from a geometric point of view, previous work has shown that the intrinsic dimension of the data could have some semantics. In this paper, we start from the consideration of this inherent topology property and propose the usage of such a semantic criterion for clustering. The corresponding learning algorithms are provided. Theoretical justification and analysis of the algorithms are shown. Promising results are reported by the experiments that generally fail with conventional clustering algorithms.

1 Introduction

Clustering [5][6] is a classical unsupervised technique. It tries to organize objects into groups whose members are similar in some way and is used to discover the natural groups in the data and to identify the hidden structures that might reside inside. It is widely used in different tasks such as data mining, computer vision, VLSI design, web page clustering and gene expression analysis, etc.

Unfortunately, clustering is not a well-defined problem. How to decide what constitutes a good clustering? Although many efforts have been devoted[5], to a large extent the problem remains *"elusive"*. It can be shown that there is no absolutely *"best"* way which suits all problems. There is still a desired need for new criteria and methods to cope with new problems.

In this paper, we consider the problem from a topology point of view, and propose the usage of the manifold dimension (or, intrinsic dimension) as a semantic criterion for clustering. The *"semantics"* comes from the facts recently discovered by the researches in nonlinear dimensionality reduction[12][10][1][4]. The primary aim of these techiques is to seek a lower dimensional embedding of a set of points, which are originally expressed in a higher dimensional space. It was discovered that the manifold dimension obtained in such a way is often associated with some semantics, which are typically correlated with highly nonlinear features of the original space.

The paper is organized as follows. In section two, we formulate the basic idea and discuss the estimation of *"local"* dimensions. In section three, based on a property of *"local"* dimensions, we give an algorithm to divide the data into a pre-assigned number (K) of clusters. An extension is further given to cope with

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 796–805, 2006.

an unknown K. After showing experiment results in section four, we summarize the paper in the final section.

2 Background

2.1 Basic Idea

In data processing, we represent an object as a collection of numbers, i.e. a vector, which specifies the different attributes' values of this object. Furthermore, the "collection of numbers also specifies the Cartesian coordinate of a point with respect to a set of axes"[11]. Therefore, any object can be identified as a point in an abstract space.

In this paper, we study the topology formed by all the points. With a hypothesis that different clusters will form different topologies, the topology properties can be used as a clustering criterion. Specifically, we consider the manifold dimension, which is an inherent topology property, as a criterion to group different clusters. A formal statement of the idea is the following.

Given n independently identically distributed observations $x_1, x_2, ..., x_n$ of a cluster in R^p. For each x_i, let's assume $x_i = g\left(y_i\right)$, where each y_i is sampled from a smooth density f in R^m ($m \leq p$) and g is an unknown function. The cluster is regarded as dominated by these m independent hidden factors. The function g is continuous and sufficiently smooth, and the neighborhood relationship is kept when mapping R^m to R^p. Then we can expect the cluster shows the topology property of an m-dimensional manifold. For different clusters that are dominated by different number of hidden factors, the analysis of manifold dimensions can be used to separate them.

2.2 Estimation of Dimension

To use the manifold dimension as a separation criterion, the intial step is to evaluate the *"local"* dimension[1] around each point. In literature, there have been several researches[9][13] on this problem. Here we use a method presented in [7], which gives a maximum likelihood estimator of the manifold dimension of the data and reaches the intrinsic dimension asymptotically.

Given a dataset $X = \{x_1, x_2, ..., x_n\}$, we hope to estimate the *"local"* dimension around each point. The basic idea of the method is to regard the density of the data, $f\left(x\right) \approx const$ in a small sphere $S_x\left(R\right)$ of radius R around data point x. The method utilizes the relationship between the *"local"* dimension d_x and the volume of the sphere $S_x\left(R\right)$. By treating the observations as a homogeneous Poisson process in $S_x\left(R\right)$, it reaches a maximum-likelihood estimator of the intrinsic dimension m:

$$\hat{m} = \frac{1}{n}\sum_{i=1}^{n} d_i \tag{1}$$

[1] Throughout this paper, we use this informal term, *"local"* dimension, to represent the dimension calculated using formula (2) or (3) around each point.

where

$$d_i = \left[\frac{1}{N(R, x_i)} \sum_{j=1}^{N(R,x_i)} \log \frac{R}{T_{ij}} \right]^{-1} \tag{2}$$

and $N(R, x_i) = \sum_{j=1}^{n} 1\{x_j \in S_{x_i}(R)\}$ is the number of points within distance R from x_i. Here, d_i may be regarded as the *"local"* dimension of a small fraction around x_i, and T_{ij} is the distance between point x_i and its j-th nearest neighbor.

When fixing the number of neighbors k for studying the Poisson process rather than using the radius of the sphere R for convenience, we get

$$d_i = \left[\frac{1}{k-1} \sum_{j=1}^{k-1} \ln \frac{T_{ik}}{T_{ij}} \right]^{-1} \tag{3}$$

3 Algorithms

With the method presented in previous section, we give an algorithm of merging the points into different clusters according to their respective *"local"* dimensions.

3.1 Basic Algorithm

First, we formulate the problem.

Problem. *Given a mixture of data points $X = \{x_1, x_2, ..., x_n\}$ from a known number*[2] *(K) clusters. The "local" dimension around each point within the same cluster is similar, while it differs for inter-cluster points. We are to use this criterion to separate the points into K clusters.*

Here we give the following algorithm:

Algorithm (Basic)

Input: *a dataset $X = \{x_1, x_2, ..., x_n\}$, the number of clusters K.*

Output: *a set of K clusters $C = \{X_1, X_2, ..., X_K\}$, with $\cup_{i=1}^{K} X_i = X$ and $X_i \cap X_j = \Phi$, $1 \le i < j \le K$.*

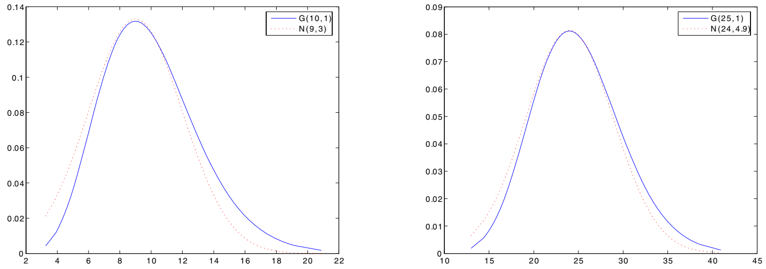
Step 1, *For each point x_i, determine its k-nearest neighbors and use formula (3) to estimate the local dimension d_i. And let $D = \{d_1, d_2, ..., d_n\}$.*

Step 2, *Model D as a Gaussian mixture. And use EM algorithm to separate D into K clusters $D_1, D_2, ..., D_K$. Each d_i is classified into D_j according to*

$$d_i \in D_j \Longleftrightarrow j = \arg_l \max p(d_i | w_l, \theta_l), \ 1 \le l \le K. \tag{4}$$

where $p(d|w_l, \theta_l)$ is the probability that the point d comes from the cluster represented by w_l. Here θ_l represent the necessary parameters for the distribution, such as the variance.

[2] We use a capital K to represent the number of clusters; while a lowercase k is used to represent the number of neighbors when studying the *"local"* dimensions.



(a) $k = 10$

(b) $k = 25$

Fig. 1. Gamma distribution (solid line) and Gaussian distribution (dotted line)

Step 3, *Separate X into K clusters $X_1, X_2, ..., X_K$ according to $D_1, D_2, ..., D_K$:*

$$\forall 1 \leq i \leq n, 1 \leq j \leq K, x_i \in X_j \Longleftrightarrow d_i \in D_j \tag{5}$$

In the second step of the algorithm, we use the expectation maximization (EM) algorithm[3][2] to separate the data into different clusters. It first assigns data points to "clusters" or density models using a "soft assignment"[8] method. Then it re-estimate the clusters or density models based on the current assignment.

3.2 Justification

For the algorithm presented above, we model D as a Gaussian mixture and use the EM algorithm to separate it, which means we have implicitly admitted the distribution of the *"local"* dimensions to be Gaussian or similar-type. In fact, this assumption can be justified by the following fact which is observed in [7]:

For each point x_i, $m^{-1} \sum_{j=1}^{k} \log \frac{T_{ik}}{T_{ij}}$ has a $Gamma\,(k, 1)$ distribution, where m is the intrinsic dimension.

Generally, we require[3] $k \geq 10$. As shown in figure 1, $Gamma\,(k, 1)$ will be very similar to a Gaussian distribution under such circumstances. This is also verified in our experiments.

3.3 Extended Algorithm

To cope with the situation where K is not known beforehand, we give an extended version of the algorithm.

The algorithm is based on a competitive learning approach, the rival penalized competitive learning (RPCL) [14]. When applied to clustering problems, the RPCL first requires a rough estimation of an upper bound (K') of K as an input. An estimation of K' is generally trivial and could be made from the previous experiences. We randomly generate these K' units, with each unit representing

[3] This is to ensure the validity of modeling the problem as a Poisson process.

the centroid of a virtual cluster. Then competition mechanisms are introduced among these units during the training. With the incoming of each data point, the values of the first winner unit are modified by a small step to adapt to the input, while the values of its rival (the second winner unit) are delearned by a smaller learning rate.

During the learning, the RPCL algorithm tries to push each winner's rival a step away from the cluster towards which the winner is moving, thus implicitly producing a force which attempts to make sure that each cluster is learned by only one weight vector. Gradually, the abundant units are eliminated.

For our problem, we propose the following algorithm which combines RPCL learning with EM algorithm.

Algorithm (RPCL EM Clustering)

Input: *a dataset* $X = \{x_1, x_2, ..., x_n\}$*, an upper bound of the number of clusters* $K^{'}$*.*

Output: *the actual number of the clusters* K*, a set of clusters* $C = \{X_1, X_2, ..., X_K\}$*.*

Step 1, *For each point* x_i*, determine its* k*-nearest neighbors and use formula (3) to estimate the local dimension* d_i*. And we get* $D = \{d_1, d_2, ..., d_n\}$*.*

Step 2, *Randomly generate* $K^{'}$ *units* $w_1, w_2, ..., w_{K'}$*;*

Step 3, *Model* D *as a Gaussian mixture. Iterate the following steps for a pre-defined number of steps:*

Step 3.1, *Randomly pick up a sample* d *from the dataset* D*, and for* $i = 1, ..., K^{'}$*, let*

$$u_i = \begin{cases} 1, \ \textit{if } i = c, \textit{ where } c = \arg_j \max \gamma_j \cdot p\left(d|w_j, \theta_j\right), \\ -1, \ \textit{if } i = r, \textit{ where } r = \arg_{j \neq c} \max \gamma_j \cdot p\left(d|w_j, \theta_j\right), \\ 0, \ \textit{otherwise.} \end{cases} \tag{6}$$

where $\gamma_j = \frac{n_j}{\sum_{i=1}^{K'} n_i}$ *and* n_i *is the cumulative number of the occurrences of* $u_i = 1$*.*

Step 3.2, *Update the weight vector* w_i *by*

$$\varDelta w_i = \begin{cases} \alpha_c\left(x - w_i\right), \ \textit{if } u_i = 1, \\ -\alpha_r\left(x - w_i\right), \ \textit{if } u_i = -1, \\ 0, \ \textit{otherwise.} \end{cases} \tag{7}$$

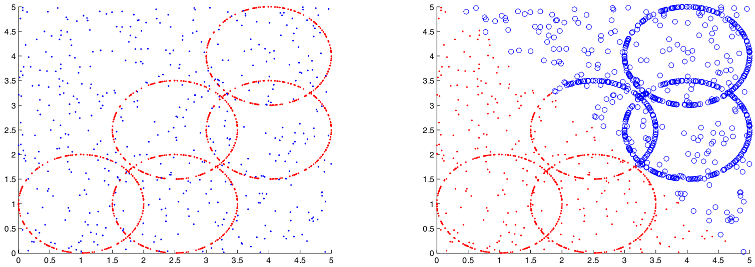
where $0 \leq \alpha_c, \alpha_r \leq 1$ *are the learning rates for the winner and rival unit, respectively. In practice, they are problem dependent and may also depend on the iteration step* t*. And it also holds* $\alpha_c \gg \alpha_r$*.*

Step 3.3, *Re-estimate the values of the parameters* $\theta_1, \theta_2, ..., \theta_{K'}$ *using EM algorithm.*

Step 4, *Separate* D *into* $K^{'}$ *clusters* $D_1, D_2, ..., D_{K'}$ *according to*

$$d_i \in D_j \Longleftrightarrow j = \arg_l \max p\left(d_i|w_l, \theta_l\right), \ 1 \leq l \leq K'. \tag{8}$$

Eliminate those D_l *if* $D_l = \varPhi$ *and re-arrange* $D_1, D_2, ..., D_{K'}$ *into* $D_1, D_2, ..., D_K$*, where* K *is the number of clusters in* D *which are not empty.*



(a) A two-cluster mixture

(b) $K-$means results

Fig. 2. A mixture of two overlapped clusters: one is sampled from five 1-d circles; the other is from a 2-d rectangle. The mixture caused complete failure to to the $K-$means algorithm. (The results are represented by different shades.)

Step 5, *Separate X into K clusters $X_1, X_2, ..., X_K$ according to $D_1, D_2, ..., D_K$:*

$$\forall 1 \leq i \leq n, 1 \leq j \leq K, x_i \in X_j \Longleftrightarrow d_i \in D_j. \tag{9}$$

The first and the last step are essentially the same as those of the basic algorithm presented in the previous section. Besides, the second and the third steps are used for the competive learning and rival penalty. In step 3.1, a new parameter γ_j is introduced for each unit w_j, and this parameter is used to ensure that all the units will have the chances to be the first winner.

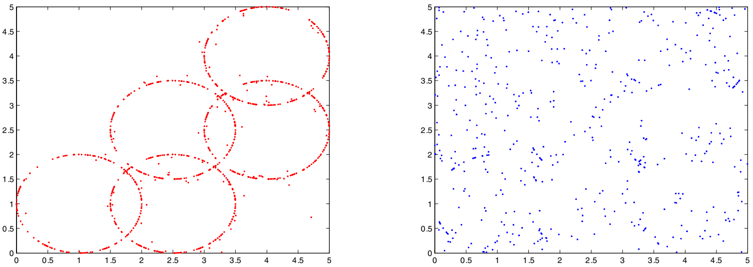
4 Experiments

In this section, we show some experiment results of the algorithms presented above. The first two experiments cope with the cases when the clusters overlap each other. Such problems will generally fail with traditional clustering algorithms. For the third experiment, the two clusters do not overlap. Although we can use other clustering methods, here we use our algorithm to show its potential to cope with clustering problems by the semantic criterion.

4.1 Experiment I

An experiment is shown in figure 2. Two clusters are completely overlapped, and we hope to recover the two clusters by their semantic meaning (five circles and one rectangle).

The results are shown in figure 3. After dimensionality analysis, most of the points are correctly clustered. Although there are several points evidently misclassified, they can be eliminated by some simple post-processings, for which we omit the discussion. However, for traditional approaches such as K-means, it fails completely.



(a) cluster 1

(b) cluster 2

Fig. 3. The clustering results after dimensionality analysis

4.2 Experiment II

The second experiment uses an example with which the number of clusters is supposed to be unknown. We artificially generate three clusters[4] overlapped in a 5-d space: the first is sampled from a 1-d curve, the second is from a 3-d manifold and the third is from a 5-d manifold. Then we run the RPCL EM algorithm.

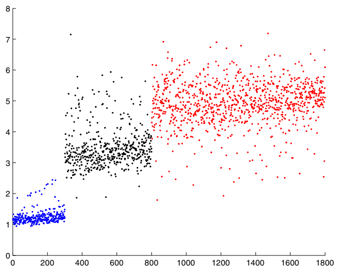


Fig. 4. The *"local"* dimensions of a mixture of three clusters: each sampled from a 1-d (300 points), 3-d (500 points) and 5-d (1000 points) manifold respectively ($k = 15$)

During the experiment, we start from $K^{'}(= 6)$ units, i.e. w_1 to w_6. And α_c and α_r are set to be 0.02 and 0.002 respectively. During the training process, w_1, w_4 and w_6 are gradually pushed away, while w_2, w_3 and w_5 get stabilized and converge to the correct centroids finally. The results are shown in figure 4 and table 1. Table 2 gives the ratio of the points correctly identified within each cluster. From the table, we can see that most of the points are correctly clustered.

[4] The $1st$ cluster is generated by: $x = \sin(t_1), y = z = r = s = t_1$. The $2nd$: $x = \sin(t_1), y = \sin(t_2), z = \sin(t_3), r = s = t_3$. The $3rd$: $x = \sin(t_1), y = \sin(t_2), z = \sin(t_3), r = \sin(t_4), s = \sin(t_5)$. The free parameters $t_1, t_2, ..., t_5$ are randomly selected within $(0, 10)$.

Table 1. Changes of the units

	Initial Value	Stabilized Value
w_1	−4.0	−7.5
$\mathbf{w}_2$	**−0.2**	**3.10**
$\mathbf{w}_3$	**6.0**	**4.92**
w_4	12.0	17.5
$\mathbf{w}_5$	**11.7**	**1.24**
w_6	0.9	20.5

Table 2. The number of points correctly clustered

	#Correct	Ratio
1-dimension (300 points)	292	97.3%
3-dimension (500 points)	430	86%
5-dimension (1000 points)	920	92%

4.3 Experiment III

In the third experiment, two image datasets are mixed (figure 5). The first is a collection of images[5] of a face, rendered with different poses and lightings; the second is a collection of images[6] of a hand, rendered with different directions.

Considering the semantics of the images, it is expected that the face images would form an abstract space with a higher intrinsic dimension than that of the space formed by the hand images. The experiment results verified our guess. After running the basic algorithm, two clusters are recovered. Table 3 shows the number of images correctly identified for each cluster under different parameter settings (also see figure 6). From the table, we can see, when the number of neighbors is appropriately chosen, all the images are correctly identified.

Table 3. Clustering results

	$k = 15$	$k = 20$	$k = 25$	$k = 30$	$k = 35$	$k = 40$
Faces (698 images)	698	698	698	698	698	698
Hands (481 images)	219	393	403	479	481	481

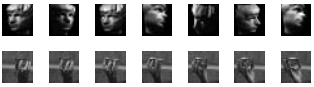
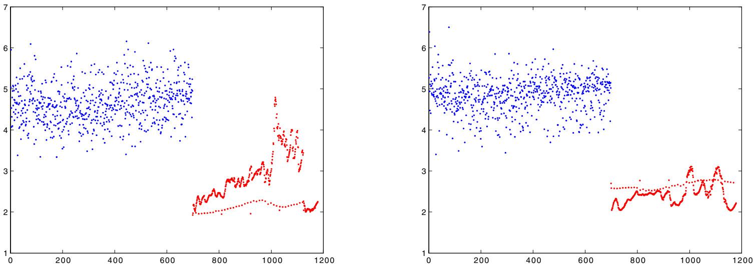


Fig. 5. Two image datasets

[5] http://isomap.stanford.edu/datasets.html

[6] http://vasc.ri.cmu.edu//idb/html/motion/hand/index.html



(a) $k = 20$

(b) $k = 35$

Fig. 6. The *"local"* dimensions around each point with different parameter settings. The first 698 points represent the face images. The rest 478 points represent the hand images.

5 Summary

In this paper, we start from the topology point of view, discuss the usage of manifold dimension as a semantic criterion for data clustering, and give the corresponding clustering algorithms. Promising results are reported on some problems that cause failures to conventional clustering algorithms. On the other hand, we still need more explorations on real datasets and test its applicability.

Acknowledgment

This research was partially supported by RGC Earmarked Grant #4173/04E and #4132/05E of Hong Kong SAR and RGC Research Grant Direct Allocation of the Chinese University of Hong Kong.

References

1. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 585-591, Cambridge, MA, 2002. MIT Press.
2. Blimes, J.A.: A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. International Computer Science Institute, UC Berkeley, 1998.
3. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Scociety B*, 39:1-39, 1977.
4. Donoho, D.L., Grimes, C.: Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. In *Proceedings of the National Academy of Arts and Sciences*, 2003.
5. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, Inc, 1988.

6. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. *ACM Computing Surveys*, 31(3):263-323, September 1999.
7. Levina, E., Bickel, P.J.: Maximum likelihood estimation of intrinsic dimension. In L.K. Saul, Y.Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 777-784. MIT Press, Cambridge, MA, 2005.
8. Michael, K., Yishay, M., Andrew, N.: An information-theoretic analysis of hard and soft assignment methods for clustering. In *Proceedings of the 13th Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, pages 282-293, San Francisco, CA, 1997. Morgan Kaufmann Publishers.
9. Pettis, K.W., Bailey, T.A., Jain, A.K., Dubes, R.C.: An intrinsic dimensionality estimator from near-neighbor information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:25-37, 1979.
10. Roweis S., Smola A.J.: Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323-2326, 2000.
11. Seung, H.S., Lee, D.D.: The manifold ways of perception. *Science*, 290:2268-2269, 2000.
12. Tenenbaum, J.B., de Silva, V., Landford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2313-2323, 2000.
13. Verveer, R., Duin, R.: An evaluation of intrinsic dimensionality estimators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):81-86, 1995.
14. Xu, L., Krzyzak, A., Oja, E.: Rival penalized competitive learning for clustering analysis, RBF net, and curve detection. *IEEE Transactions on Neural Networks*, 4, 1993.

Improved Clustering and Anisotropic Gradient Descent Algorithm for Compact RBF Network*

Delu Zeng, Shengli Xie, and Zhiheng Zhou

College of Electronic & Information Engineering, South China University of Technology
510641, Guangzhou, China
Donald_zeng@163.com

Abstract. In the formulation of radial basis function (RBF) network, there are three factors mainly considered, i.e., centers, widths, and weights, which significantly affect the performance of the network. Within thus three factors, the placement of centers is proved theoretically and practically to be critical. In order to obtain a compact network, this paper presents an improved clustering (IC) scheme to obtain the location of the centers. What is more, since the location of the corresponding widths does affect the performance of the networks, a learning algorithms referred to as anisotropic gradient descent (AGD) method for designing the widths is presented as well. In the context of this paper, the conventional gradient descent method for learning the weights of the networks is combined with that of the widths to form an array of couple recursive equations. The implementation of the proposed algorithm shows that it is as efficient and practical as GGAP-RBF.

1 Introduction

Radial Basis Function (RBF) networks, due to their simple topological structures while retaining outstanding ability of approximation, are being used widely in function approximation, pattern recognition, and time series prediction.

Generally there are several crucial factors which seriously affect the performance of the RBF networks, i.e., the number of the hidden neurons, the center and the width for each neuron, and the weights. The original RBF networks [1] require that there be as many neurons as the observations (inputs).Thus they bring on high computational cost particularly for the case of bulky observations. In order to reduce the number of the hidden neurons, some compact RBF networks have been proposed [2]-[5].

However, among the above mentioned factors, the choice of the centers has the most critical effect on the performance of the network and plenty of study has been done on the choice of centers [2]-[7].

The algorithms proposed by S.Chen [2], [5] obtain a more compact network by orthogonal least square (OLS) method. The scheme has educed the contribution to the output variance from each neuron.

* The work is supported by the National Natural Science Foundation of China for Excellent Youth (Grant 60325310), the Guangdong Province Science Foundation for Program of Research Team (Grant 04205783), the Specialized Prophasic Basic Research Projects of Ministry of Science and Technology, China (Grant 2005CCA04100).

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 806 – 813, 2006.

Integrated with forward subset selection and 0th-order regularization, Orr [3] presented a regularized forward selection (RFS) algorithm for RBF networks. The algorithm used only one preset parameter, the basis function width.

To further study on the influence on the error by the locations of centers, Panchapakesan [6] proposed a new result on the bounds for the gradient and Hessian of the error considered as a function the centers, the widths, and the weights, for justification of moving the centers.

G. Huang [4] proposed a generalized growing and pruning RBF (GGAP-RBF) network. In their paper, a definition of "significance" is provided to judge the significance of the hidden neurons. Using this definition to grow or prune the hidden neurons one can establish a parsimonious RBF network with most significant ones. It functions well and it seems the GGAP-RBF is a great theoretical breakthrough on establishing a compact RBF network.

This paper looks into the problem of learning the centers with an improved clustering (IC) scheme and the widths with an anisotropic gradient descent (AGD) method.

2 RBF Network

The validity of RBF network is guaranteed by the theory of Reproducing Kernel Hilbert Space (RKHS), in which the dot product is computed by the kernels. In the field of RBF network one use series of RBFs to play the role of kernels as in RKHS.

Let $X = \{x_{1,}x_{2,} \cdots, x_N\}$, where $x_i = (x_{i1}, x_{i2}, \cdots x_{il}) \in R^l$, be the observations, and $Y = \{y_{1,}y_{2,} \cdots, y_N\}$ be the corresponding desired outputs. Without loss of generality, the Gaussian $g(\cdot) = \exp(-\frac{\|\cdot\|^2}{2\sigma^2})$ is usually chosen to take the role of RBF.

Then output of the system is:

$$f(X, C, \sigma, W) = \sum_j w_j \exp(-\frac{\| x_i - c_j \|^2}{2{\sigma_j}^2}), \tag{1}$$

where $C = \{c_1, c_2, \cdots, c_M\}$ and $\sigma = \{\sigma_1, \sigma_2, \cdots, \sigma_M\}$ are centers and widths of the hidden neurons, respectively, $\|\cdot\|$ is the Euclidean norm, and $W = (w_1, w_2 \cdots, w_M)$ are the weights connecting the hidden neurons with the output.

3 Proposed IC-AGD Algorithm

Besides learning the centers, we study how to design the corresponding widths as well since they do affect the performance of the RBF networks either and can not be neglected.

In the section, an improved clustering (IC) scheme for locating the centers is described first, followed by the anisotropic gradient descent (AGD) method to decide the relative widths.

3.1 An Improved Clustering Scheme

Given a set of distinct observations $\{x_i\}_{i=1}^{N}$, where $x_i \in R^l \quad (i=1,\cdots N)$, we need to cluster the observations into some categories without any a priori information of the number of category. A proposed scheme for efficiently clustering the observations is formulated as follows:

A. Compute the mean position point $\bar{x}$ for all the observations as follows:

$$\bar{x} = \frac{1}{N}\sum_{i}^{N} x_i \,. \tag{2}$$

B. Compute the distance d_i between each of the observations $\{x_i\}_{i=1}^{N}$ and the mean position point $\bar{x}$, i.e., $d_i^2 = \| x_i - \bar{x} \|^2$. Then rank $\{d_i\}_{i=1}^{N}$ from small to large by quit sort scheme. Without loss of generality, they are still denoted by $d_1 \le d_2 \le \cdots \le d_N$.

C. Define

$$d_{\min} \triangleq \min\{d_i, i=1,\cdots,N\} = d_1, \tag{3}$$

and

$$d_{\max} \triangleq 2\max\{d_i, i=1,\cdots,N\} = 2d_N \,. \tag{4}$$

Let

$$d \triangleq \lambda \cdot d_{\min} + (1-\lambda)\cdot d_{\max} = \lambda \cdot d_1 + 2(1-\lambda)\cdot d_N \,, \tag{5}$$

where $\lambda \in [0,1]$ is a preset parameter.

D. Set a relationship matrix $\{r_{i,j}\}_{i,j=1}^{N}$, which intends to indicate the cluster membership of each point from the observations, namely a Cluster Indication Matrix (CIM). And initialize the CIM as a zero matrix.

E. In order not to calculate all of the distances between every two points, we firstly intend to find a coarse CIM for the observations.

For each observation $x_i \in X = \{x_i\}_{i=1}^{N}$, do the following steps from $i=1$ to N :

a. Denote $O(x_i)$ the neighborhood centered at x_i and with radius d .

b. Compute the distance between x_i and x_j , if x_j satisfies that $r_{ij}=0$ and $x_j \in X_1 = \{x_j \mid \; | d_j - d_i | < d \;\; and \;\; i \ne j\} \subset X$ (Fig.1). The way that we

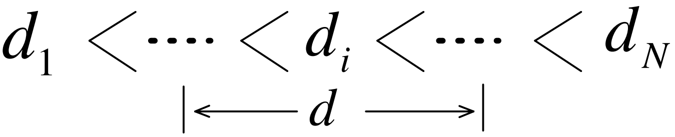

$d_1 < \cdots < d_i < \cdots < d_N$
$|\longleftarrow d \longrightarrow|$

Fig. 1. Search the points in every neighborhood

search the points within the subset X_1 of X would greatly lower the computation complexity.

c. Then judge whether the observation x_j is in $O(x_i)$, and let

$$r_{ij} = r_{ji} = \begin{cases} |x_i - x_j|, & if \quad x_j \in O(x_i) \\ -1 & ,else \end{cases}, \tag{6}$$

where x_j is a suspicious neighborhood point of x_i when $r_{ij} = -1$.

F. For the partition obtained from step **E** will end up with intersections between some coarse clusters, It is necessary to refine the CIM to make sure the **rule** is guaranteed, which i and k should belong to the same cluster, if i and j, j and k belong to the same cluster respectively. Thus the suspicion for membership of certain points is eliminated .To tackle this, we refine the CIM in the following way:

$$\text{If} \begin{cases} r_{ij} > 0 \\ r_{jk} > 0 \\ r_{ik} = 0 \end{cases} \text{or} \begin{cases} r_{ij} > 0 \\ r_{jk} > 0 \\ r_{ik} = -1 \end{cases} \text{, then set } r_{ki} = r_{ik} = 1. \tag{7}$$

From the refined CIM, we know the observations $\{x_i\}_{i=1}^{N}$ can be partition into M clusters, namely $\{Q_1, Q_2, \cdots, Q_M\}$ and it is obvious that the lower bound for the distance of any two clusters is d .

3.2 Center Locating and Coarse Width Setting

Let

$$c_i = \frac{1}{|Q_i|}\sum_{j=1}^{|Q_i|} x_j \text{, and } \sigma_i^2 = \frac{1}{|Q_i|}\sum_{j=1}^{|Q_i|} \| x_j - c_i \|^2 \tag{8}$$

where $x_j \in Q_i$ and $i = 1, \cdots, M$.

3.3 Widths and Weights Learning

Many papers have focused on locating the centers as well as the weights, while it is noted that the designing of the widths is still important. In this part, we formulate an anisotropic gradient descent (AGD) method. According to the distribution of the observations in R^l, the change of the observations mainly processed by those RBFs may vary between directions. Our AGD method is to update the widths of the kernels with a width scaling factor for the above coarse widths.

Let us define the refined widths of the RBF:

$$\sigma_*^2 = (s_1\sigma_1^2, s_2\sigma_2^2, \cdots, s_M\sigma_M^2), \tag{9}$$

where $S = (s_1, s_2, \cdots, s_M)$ is a scaling vector and each of its components is positive.

Then the mean square error (MSE) of the system is switched to a function of S and W, i.e.:

$$E(\sigma_*, W) = \frac{1}{N}\sum_{i=1}^{N} \| Y - f(X, C, \sigma_*, W) \|^2$$

$$\Rightarrow \quad E(S, W) = \frac{1}{N}\sum_{i=1}^{N} (y_i - f(X, C, S, W))^2 \ . \tag{10}$$

Replace f with Gaussian radial function, and differentiate the equation (10) with respect to w_j and s_j, $j = 1, \cdots M$, respectively, then we have:

$$\frac{\partial E(S,W)}{\partial w_j} = 2\frac{\partial \gamma(S,W)}{\partial w_j}\gamma^T(S,W) = -2G(X, s_j)\gamma^T(S,W), \tag{11}$$

$$\frac{\partial E(S,W)}{\partial s_j} = 2\frac{\partial \gamma(S,W)}{\partial s_j}\gamma^T(S,W) = -2w_j\frac{\partial G(X, s_j)}{\partial s_j}\gamma^T(S,W), \tag{12}$$

where $\gamma(S,W) = (\gamma_i(S,W))_{i=1}^{N}$, $\gamma_i(S,W) = y_i - \sum_{j=1}^{M} w_j g(x_i, s_j)$, and $G(X, s_j) = (g(x_i, s_j))_{i=1}^{N}$, $g(x_i, s_j) = \exp(-\frac{\| x_i - c_j \|^2}{2 s_j \sigma_j})$, $i = 1, \cdots, N$.

Apply the gradient descent for $E(S, W)$ with respect to S and W respectively, we obtain the coupled recursive equations for S and W as follows:

$$\begin{cases} w_j(n+1) = w_j(n) + 2\eta_1 G(X, s_j(n))\gamma^T(S(n), W(n)) \\ s_j(n+1) = s_j(n) + 2\eta_2 w_j(n)\dfrac{\partial G(X, s_j(n))}{\partial s_j}\gamma^T(S(n), W(n)) \end{cases}, \tag{13}$$

where η_1 , η_2 are two different learning steps with non-negative value and $j = 1, \cdots M$.

4 Implementation

In this section, an implementation of our algorithm is applied to a time series problems in function approximation area.

The chaotic Markey-Glass time series prediction is a well known benchmark prediction problem which is generated from the following delay differential equation:

$$\begin{cases} \frac{du(t)}{dt} = -0.1u(t) + \frac{0.2u(t-17)}{1+u^{10}(t-17)} \\ u(t-17) = 0.3 \end{cases} . \tag{14}$$

Table 1. Performance Comparisons between IC-AGD and GGAP-RBF

Algorithms	CPU Time (s)	Training RMSE	Testing RMSE	Number of neurons
GGAP(2-norm)	9.432	0.031342	0.058692	10
IC-AGD	8.198	0.011465	0.062481	9

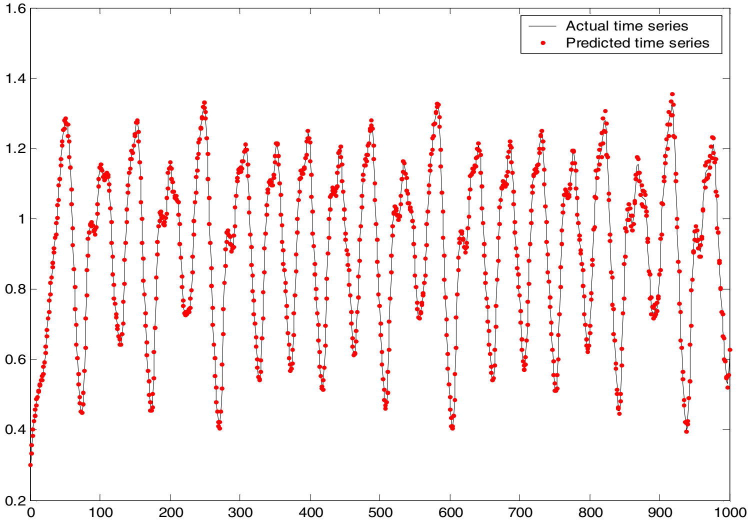


Fig. 2. The predicted with the IC-AGD and actual time series

Resample $N = 1000$ points from the above equation according to 1 sample period. With 20 sample steps ahead, we aim to predict the value of $u(t+20)$ by the values $\{u(1), u(2), \cdots, u(t)\}$. Let $\{x_i = (u_{i-20}, u_{i-20-6,} u_{i-20-12,} u_{i-20-18})^T\}_{i=1}^{1000}$ be the multiple inputs for the RBF network, while $\{y_i\}_{i=1}^{1000}$ be the corresponding outputs, where the first 800 pairs are for training, and the last 200 pairs are for testing. We have the following results:

Table 1 gives an comparison of the performance between the proposed algorithm and GGAP-RBF algorithm[4], where the parameters of GGAP is $e_{\min} = 0.01$, $\kappa = 0.85$, $\varepsilon_{\max} = 0.7$, $\varepsilon_{\min} = 0.07$, and the parameter in the proposed algorithms is $\lambda = 0.5$. Fig.2 shows the performance for the approximation ability of the proposed algorithm for the above time series prediction problems, where the red points denote the predicted time series, and the black curve denotes the actual time series.

The simulations show that the proposed algorithm performs as efficient as the GGAP-RBF, and will be practical in function approximation area either.

5 Conclusions

In this paper, an efficient algorithm IC-AGD for establishing a compact RBF network is presented. This algorithm consists of an improved clustering (IC) scheme for placing the centers, and an anisotropic gradient descent (AGD) method for learning the widths combined with learning the weights by conventional gradient descent. In the IC scheme, we take advantage of the relation between the observations and the mean position point to lower the computational complexity. And in the AGD method, we quote a scaling vector for the widths since the clusters may vary between directions.

Note that in the process of IC scheme for locating centers, the value of λ for threshold d will be much more favorable when it is optimized from maximizing the distance between clusters.

In conclusion, implementation shows that the proposed IC-AGD algorithm is as efficient and practical as the newly presented GGAP-RBF [4]. What is more, the proposed IC scheme can be utilized for other clustering problem.

References

1. Broomhead, D.S., Lowe, D.: Multivariable functional interpolation and adaptive networks. Complex Syst.,vol.2, (1988) 321-255
2. Chen, S., Cowan, C. F. N., Grant, P. M.: Orthogonal least squares learning algorithm for radial basis function networks. IEEE Trans. Neural Netw., vol. 2, no. 2, (1991) 302–309
3. Orr, M. J. L.: Regularization on the selection of radial basis function centers. Neural Computat., vol. 7, (1995) 606–623
4. Huang, G. B., Saratchandran, P., Sundararajan, N.: A Generalized Growing and Pruning RBF(GGAP-RBF) Neural Network for Function Approximation. IEEE Trans. Neural network, vol.16, no. 1, (2005) 57-67

5. Chen, S., Chng, E. S., Alkadhimi, K.: Regularized orthogonal least squares algorithm for constructing radial basis function networks. Int. J. Control, vol. 64, no. 5, (1996) 829–837
6. Panchapakesan, C., Palaniswami, M., Manzie. C.: Effects of moving the centers in an RBF network. IEEE Trans. Neural Network, vol.13, No.6, (2002) 1299-1307.
7. Xie, S.L., He, Z.S., Gao, Y.: Adaptive Theory of Signal Processing. 1st ed. Chinese Science Press, Beijing (2006).

Clustering Massive High Dimensional Data with Dynamic Feature Maps

Rasika Amarasiri[1], Damminda Alahakoon[1], and Kate Smith-Miles[2]

[1] Clayton School of Information Technology,
Monash University, VIC, 3800, Australia
[2] School of Engineering and Information Technology,
Deakin University, Burwood, VIC, 3125, Australia
(Rasika.Amarasiri, Damminda.Alahakoon)@infotech.monash.edu.au,
katesm@deakin.edu.au

Abstract. This paper presents an algorithm based on the Growing Self Organizing Map (GSOM) called the High Dimensional Growing Self Organizing Map with Randomness (HDGSOMr) that can cluster massive high dimensional data efficiently. The original GSOM algorithm is altered to accommodate for the issues related to massive high dimensional data. These modifications are presented in detail with experimental results of a massive real-world dataset.

1 Introduction

The Self Organizing Map (SOM) [1] was developed in the early 80s by Professor Teuvo Kohonen. The ability of the SOM algorithm to present the data in a topologically ordered map has made it popular in many data mining applications [2]. The basic SOM algorithm just like any other new algorithm had many flaws in it. Addressing these flaws emerged into the whole range of algorithms commonly known as feature map algorithms. These algorithms address issues such as the initialization of the SOM [3], determination of the ideal height and width of the map [4-8], faster convergence [9, 10] and application to large datasets [11].

Many growing variants of the SOM were introduced to address the issue of determining the ideal height and width of the SOM. One such algorithm was the Growing Self Organizing Map (GSOM) [4]. This paper presents several modifications to the GSOM algorithm to overcome some difficulties faced by the GSOM when applied to massive, very high dimensional datasets such as text and web document collections. The modifications are presented as a new algorithm called the High Dimensional Growing Self Organizing Map with Randomness (HDGSOM*r*) [12, 13]. This paper gives prominence to the details of the modifications as these have not been published earlier. The paper also presents results of the largest real-world dataset processed with the algorithm that exceeds over one million records.

The paper is organized as follows: Section 2 of the paper presents the GSOM algorithm in brief. The issues addressed by the HDGSOMr algorithm are presented in Section 3 and the HDGSOMr algorithm is described in Section 4. Experimental

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 814 – 823, 2006.

results obtained by the HDGSOMr on a massive real-world dataset are presented in Section 5. The conclusions and some future work are highlighted in Section 6.

2 The Growing Self Organizing Map

The Growing Self Organizing Map (GSOM) [4] was developed as a solution to the problem of identifying a suitable height and width for the SOM for a given dataset. It starts with a minimal number of nodes (usually 4) and grows from the boundary to accommodate for the clusters in a given dataset. New nodes are grown when the accumulated error in a boundary node exceeds a pre-defined threshold called the Growth Threshold (*GT*). The *GT* is calculated based on the dimension of the dataset and a parameter called the Spread Factor (*SF*) that can be used to control the spread of the map. Due to space limitations the readers are encouraged to read [4] for further details of the GSOM algorithm.

The GSOM had mainly been used with data that was having dimensions less than about 100 and less than about 1000 inputs. It was interesting to see if the GSOM could be used for massive datasets with very large dimensions (>1000). An attempt was made to apply the GSOM in clustering web page content of a large collection.

When this dataset was presented to the GSOM it failed to generate any new nodes even at very large *SF* values as the *GT* calculated was much larger than the accumulated errors that are generated over the whole life cycle of the GSOM. In order to facilitate the growth of nodes, the *GT* calculation was modified as $GT = -\ln(D) \times \ln(SF)$ which reduced the *GT* significantly. This modification enabled the clustering of text and other large dimensional data using the GSOM.

Although the clustering was possible, the growth of the network was not perfect. The main problems that were identified were:

1. The growing of new nodes was only from one initial node although the nodes were spread around the other initial nodes in a spiral effect.
2. Some of the branches that had grown merged with other branches when forming the clusters as the map had actually grown like a spiral.
3. Excessive amounts of non-hit nodes (nodes that did not have any inputs mapped on to them) spread in many parts of the map.

These abnormalities were identified to be caused by several reasons. The root of all these problems was the noticeably large errors due to the large dimensions. The maximum possible error in distance calculations is equal to the dimension of the input vectors (*D*) if the weight vector of the node consists of all 0 and the input vector consists of all 1s or vice versa. When *D* becomes very large, the maximum and therefore the average errors recorded increase extensively.

3 Addressing the Issues of High Dimensions and Massive Data

To address the abovementioned issues several modifications were proposed and evaluated. These include:

1. Preceding the growing phase by a calibration phase and initializing the first nodes based on the calibration.
2. Multi-node initialization where the initial map contains more than 4 nodes.
3. A modified growing phase that starts with a high *GT* value and then gradually reducing to a value decided by the calibration phase or otherwise.
4. Preventing new node growth from immature nodes until they have had sufficient time to adapt to a good weight vector.
5. Introduction of smoothing phases in between growing.
6. Higher learning rates during interleaved smoothing.
7. A low neighborhood kernel during the growing phase.
8. Breaking of the neighborhood effect based on a distance metric.
9. Growth Thresholds calculated based on sparse dimensions.
10. Modification of the weight updation with Random noise to exit from local minima.

Out of these proposed modifications only methods 3, 4, 5, 6, 7, 9 and 10 were recommended as the others had negative impacts. These modifications are briefly discussed in this Section.

3.1 A Modified Growing Phase That Starts with a High GT Value

One of the reasons for the original GSOM to grow only from one initial node and spiral around was because the modified *GT* was too low. The unordered state of the map caused very large errors to be accumulated in the nodes. The reduced number of nodes in the initial stages means that the total errors are distributed among a few nodes causing the *GT* to be exceeded within a few inputs.

In order to control the growth of new nodes it was proposed to have a higher *GT* value initially and gradually reduce it as the map grows. This allows time for the existing nodes to get stabilized before growing new nodes. In order to keep the algorithm simple a uniform stepwise reduction of the *GT* was proposed.

Two *GT* values were calculated based on the dimension of the dataset as the initial *GT* value (*GT*1) and the final *GT* value (*GT*2). *GT*1 was a higher value and was calculated as $GT1 = -D \times \ln(SF)$ which was the GT calculation for the original GSOM. The lower value was calculated as $GT2 = -\ln(D) \times \ln(SF)$ which was the modified GT value used in the initial experiments.

3.2 Preventing the Growth of New Nodes from Immature Nodes

This modification was proposed to further stabilize the map from growing unwanted nodes. In the original GSOM algorithm, new nodes are generated as soon as the accumulated error in a boundary node exceeds the GT value. But since many of the boundary nodes have not been in the map for a longer time, they may not have had sufficient time to adapt their weights. When new nodes are grown from such unstable nodes, the newer nodes become unstable causing the entire map also to be unstable.

To prevent this it was proposed to allow the new nodes to be aged using a constraint before they could start growing new nodes. Each new node had to wait till an entire round of the input dataset was presented to the map so that it would have adapted itself to suit the entire dataset rather than a fraction of it.

3.3 Interleaved Growing and Smoothing Phases

In order to further stabilize the newly grown nodes it was proposed to allow them more time to self organize. This was achieved by introducing several smoothing rounds in between each growing round. The smoothing rounds had the same neighborhood kernels as the preceding growing round, but did not allow new node growth.

The extra time given for the nodes to self organize decreased the number of unwanted nodes as the accumulated errors in the new nodes were much less than when they were initialized with weight vectors calculated from unstable boundary nodes.

3.4 Higher Learning Rates During Interleaved Smoothing Phases

To give the interleaved smoothing a boost in putting the existing nodes in place, the learning rate over the interleaved smoothing phases in the growing phase was increased. The increased learning rate enabled the nodes to quickly move into position and reduce the time needed for better positioning. The increased learning meant that there would be excessive movements in the nodes during this time that might introduce higher errors to the map, but the lower learning rate during the growing phases allowed the nodes to settle down and the final smoothing phases took away the remaining creases in the map.

The increased learning rates during the interleaved smoothing phases allowed bringing the number of interleaved smoothing iterations down to two. Even one interleaved iteration was sufficient to produce a reasonably smooth map, but it was decided to have two interleaved smoothing iterations to ensure better quality. The second interleaved smoothing phase had a slightly lower learning rate than the first one that ensured convergence.

3.5 Low Neighborhood Size During Growing Phase

The neighborhood of the Self Organizing Map (SOM) algorithm is the heart of producing a properly organized feature map. Typically the SOM involves a very large neighborhood kernel during its initial stages. The recommended neighborhood kernel size is about half of the map width [2]. However, the computational intensity of a very large neighborhood kernel is very high. This is because for each weight adaptation of a winner node, a large number of its neighbors also get their weights adapted.

The larger neighborhood kernel is used in order to eliminate twists in the map and ensure proper organizing of the map. When the map size is smaller, the required neighborhood kernel is also smaller as the recommended size is half the map size. It has also been noted that maps smaller than 5 x 5 are more robust to twisting than larger maps. In these maps the neighborhood kernel can be as low as 2 during the initial stages [2].

The Growing Self Organizing Map (GSOM) had been using lower neighborhood kernels during its growing phase because of its smaller initial map size of 2 x 2. However, the typical neighborhood size of the GSOM's growing phase was typically around 2-3. With the extended smoothing phases introduced into the proposed algorithm, it was identified that the map is more organized from the initial stages and a larger neighborhood kernel was not that important. If the neighborhood kernel could be reduced, it would add value to the algorithm as it was intended for use on massive datasets. The reduction of the neighborhood kernel by even 1 would payoff significantly with the massive dataset size.

Experiments were carried out with smaller neighborhood kernels and it was noted that even a neighborhood kernel of 1 during the growing and interleaved smoothing phases was sufficient to produce a map that could be smoothed easily with a neighborhood kernel of 2 and 1 in the final smoothing phases.

3.6 Growth Thresholds Based on Sparse Dimensions

The modified algorithm of the GSOM was to be used mainly on text data. One of the important characteristics of textual data was that although the final dimension of the data set was very large, the records were very sparse due to the distribution of words in the text corpus. In many of the records the actual non-zero dimensions was less than 10% of the full dimension of the data set. Since the GSOM was using the dimension of the dataset in the calculation of the growth thresholds, the use of the total dimension was an over kill. A more representative measure of the dimension was much more suitable.

Two options were selected as the more representative dimensionality parameter. These were the maximum non-zero dimension and the average plus two standard deviations of the non-zero dimensions. Since the value for the dimensions is now lower, the resulting growth threshold value was a bit lower. In order to have a sufficiently higher growth threshold to get back the control of the growing, the growth threshold values were multiplied by a constant.

The new growth threshold calculation formulas are: $GT1 = -D \times \ln(SF) \times k$ and $GT2 = -\ln(D) \times \ln(SF) \times k$ where k is a constant called the *multiplication factor*. After several experiments over the 75 datasets, the value for k was decided to be 50 for a SF of 0.1. Over the experiments it was noted that k had a much more subtle control over the spread of the map than SF. Larger k values resulted in smaller maps while smaller values resulted in larger maps.

3.7 Modification of the Weight Updation with Random Noise

Randomness [14] is a phenomenon very frequently used in many algorithms including the Self Organizing Map (SOM). It has been very successfully used in Genetic Algorithms (GA) [15], simulated annealing [16] and neural networks.

Random numbers are used in initializing the codebook vectors of the SOM. The inputs to the SOM are presented in a random order to prevent the order of the inputs influencing the convergence of the map [2]. This process requires a heavy input/output load on the system when very large datasets are processed as the inputs will be selected in random from different parts of the dataset. If the data could still be presented in the same natural order they are stored in the storage, but some effects can be introduced to simulate the random presentation of the inputs, it would payback in reducing the processing overhead of the algorithm.

A novel mechanism of updating the weight vectors of the winner nodes using some random values was tested for this purpose. This had never been done before on any feature map algorithm. The proposed new weight updation formula is: $w_i^{new} = w_i^{old} + [\alpha + (r - 0.5) \times \alpha \times 2](x_i - w_i^{old})$ where w_i^{new} is the updated weight of the i^{th} component of the weight vector of the node, w_i^{old} is the weight of the i^{th} component of the weight vector before updation, x_i is the value of the i^{th} component of the input, α is the learning rate and r is a random number in the range of 0 and 1. The equation modifies the learning rate α by increasing or decreasing it by a fraction of itself.

The modified weight updation formula replaces all weight updation formulae in the algorithm. This will enable the smaller learning rates in the final smoothing phases to take away any unwanted changes introduced by the random weight adjustments.

4 The High Dimensional Growing Self Organizing Map with Randomness (HDGSOMr)

The algorithm incorporating all of the proposed modifications to the GSOM was named the High Dimensional Growing Self Organizing Map with Randomness (HDGSOMr).

The HDGSOM algorithm consists of three phases:

1. An Initialization Phase
2. A growing phase that has interleaved smoothing phases and
3. Two smoothing phases

As the HDGSOMr is based on the GSOM, it is created with 4 nodes connected to each other in a rectangular shape and initialized with random weight vectors. The initialization phase of the HDGSOMr is different from the GSOM in the calculations of the Growth Thresholds.

The initialization phase of the HDGSOM is as follows:

```
Initialize the 4 initial nodes with random weight
vectors.

Calculate the average and standard deviation of
non-zero dimensions.

Calculate the Growth Thresholds.
```

The growing phase of the HDGOSM is different from the GSOM in many ways. To start with, it has varying growth thresholds in each epoch. The growth thresholds are decreased from *GT*1 to *GT*2 in equal steps as described in Section 3.1. Each growing epoch is interleaved with multiple smoothing epochs as indicated in Section 3.3.

The growing phase can be summarized as follows:

```
Calculate the step value of the Growth Thresholds
For each growing round
    For each input
        Present input and find winner node
        Calculate error and accumulate error
        If error exceeds GT and winner is a boundary
        node and winner is mature
            Grow new nodes from winner
        End If
        Adapt Weights of winner and neighbors
    End For
    For number of interleaved rounds
        For each input
            Present input and find the winner $j$
            Adapt Weights of $j$ and neighbors
    End For
Increment GT
End For
```

The growing phase is followed by two smoothing phases that are almost similar to the smoothing phases of the SOM. These two phases have diminishing learning rates that smooth out the map to produce crispier clusters.

The smoothing phases can be summarized as follows:

```
For each smoothing round
    For each input
        Present input to network and find winner node
        Adapt Weights of winner and neighbors
    End For
End For
```

Typically the number of rounds is set to be 10 and 5 for smoothing phase 1 and 2 respectively.

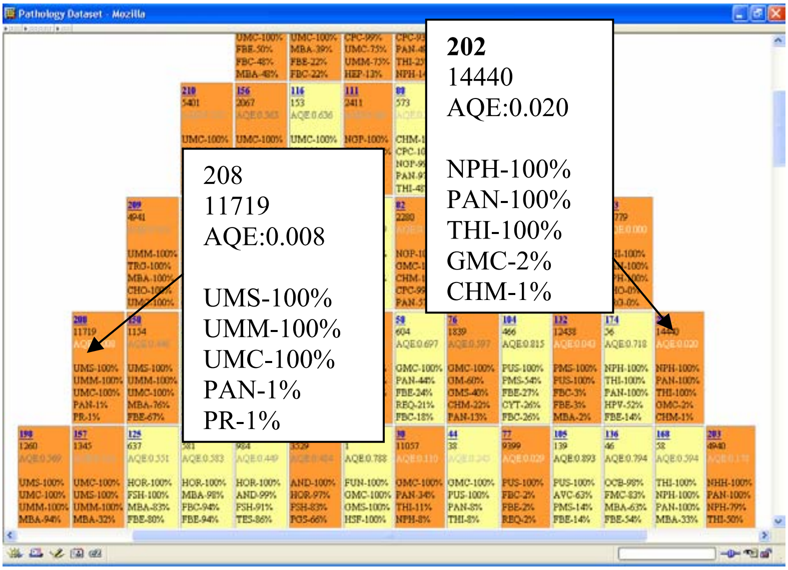
The weight adaptation rules of the HDGSOMr are differing from that of the Self Organizing Map because of the random noise component. $w_i^{new} = w_i^{old} + [\alpha + (r - 0.5) \times \alpha \times 2] \times \eta \times (x_i - w_i^{old})$ where w_i^{new} is the updated weight of the i^{th} component of the weight vector of the node, w_i^{old} is the weight of the i^{th} component of the weight vector before updation, x_i is the value of the i^{th} component of the input, α is the learning rate and r is a random number in the range of 0 and 1. η is the neighborhood kernel contribution to the weight.

5 Experimental Results

The HDGSOMr algorithm was tested with many benchmark text and web datasets as well as several real-world datasets. All of these different sized datasets were possible to be clustered within 45 iterations of the HDGSOMr compared to several hundred iterations in the SOM. The incremental growing and smoothing of the map allows it to be organized from the early stages and achieve a well-organized map within this short time period.

Several experimental results from clustering benchmark datasets are presented in [12, 13]. An online analytical tool developed based on the HDGSOMr algorithm and the results of clustering the autopsy reports of all deaths from 1995-2004 in the state of Victoria in Australia is presented in [17]. Due to space limitations only the details of the largest dataset processed up to date is presented in this paper.

The largest datasets that was clustered using the HDGSOMr was a collection of clinical notes and associated tests with the pathology requests in Australia. The dataset consisted of over 1,200,000 records with a dimension of over 800 words and over 900 tests. Both the tests and the clinical notes were clustered independently.



208
11719
AQE:0.008

UMS-100%
UMM-100%
UMC-100%
PAN-1%
PR-1%

202
14440
AQE:0.020

NPH-100%
PAN-100%
THI-100%
GMC-2%
CHM-1%

Fig. 1. A section of the map produced from the test codes in the pathology dataset

The resulting map from the clinical notes had 220 nodes and the map from the test codes had 218 nodes. A subset of the online map produced from the test codes dataset is illustrated in Fig.1. Two of the nodes content are highlighted. The first number in boldface is the node number. The next line contains the number of inputs mapped to that node and the average quantization error for that node is given in the third line. The next 5 lines contain the most dominant tests within that node with their respective percentages. In the clinical notes data set the last 5 lines contained the most dominant words within the node.

6 Conclusions and Future Work

The paper presented several modifications that were proposed to overcome the problems faced by the GSOM when applied to massive very high dimensional datasets. The resulting algorithm named HDGSOMr was capable of clustering all the benchmark and real-world datasets presented to it within a fixed 45 iterations. The random weight updation proposed in Section 3.7 has enabled the algorithm to cluster these datasets efficiently within such a short time frame while having the inputs presented to it in sequential order.

The proposed reduction in neighborhood size has further reduced the processing time required in processing massive datasets. The clustering of a massive dataset of over 1 million records has proven the ability of the algorithm.

Currently the HDGSOMr algorithm is being applied as a single level map. Work is being carried out to implement the algorithm as a hierarchical map generation tool where nodes with very large number of inputs mapped will be automatically generated into a detailed map.

A batch version of the HDGSOMr is also being evaluated to enable the algorithm to utilize the power of grid and parallel computers. Although a batch version of the SOM [10] is available, the growing of the map in the GSOM and the HDGSOMr causing the map size to change over subsequent rounds does not allow the algorithm to utilize this method. The results of these experiments will be available in future publications.

References

1. Kohonen, T., *Self Organized formation of Topological Correct Feature Maps.* Biological Cybernetics, 1982. **43**: p. 59-69.
2. Kohonen, T., *Self Organizing Maps.* Third ed. 2001: Springer.
3. Su, M.-C., T.-K. Liu, and H.-T. Chang, *Improving the Self-Organizing Feature Map Algorithm Using an Efficient Initialization Scheme.* Tamkang Journal of Science and Engineering, 2002. **5**(1): p. 35-48.
4. Alahakoon, D., S.K. Halgamuge, and B. Sirinivasan, *Dynamic Self Organizing Maps With Controlled Growth for Knowledge Discovery.* IEEE Transactions on Neural Networks, Special Issue on Knowledge Discovery and Data Mining, 2000. **11**(3): p. 601-614.
5. Fritzke, B., *Growing cell structures – a self-organizing network for unsupervised and supervised learning.* Neural Networks, 1994. **7**(9): p. 1441-1460.

6. Fritzke, B., *Growing grid-a self-organizing network with constant neighborhood range and adaptation strength.* Neural Processing Letters, 1995. **2**(5): p. 9-13.
7. Fritzke, B., *A growing neural gas network learns topologies*, in *Advances in Neural Information Processing Systems 7,*, G. Tesauro, D.S. Touretzky, and T.K. Leen, Editors. 1995, MIT Press: Cambridge MA. p. 625-632.
8. Rauber, A., D. Merkl, and M. Dittenbach, *The Growing Hierarchical Self-Organizing Map: Exploratory Analysis of High Dimensional Data.* IEEE Transactions on Neural Networks, 2002. **13**(6): p. 1331-1341.
9. Kaski, S. *Fast winner search for SOM-based monitoring and retrieval of high-dimensional data*. in *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*. 1999: IEE.
10. Kohonen, T., *Fast Evolutionary Learning with Batch-Type Self-Organizing Maps.* Neural Processing Letters, 1999. **9**(2): p. 153-162.
11. Kaski, S., et al., *WEBSOM- Self Organizing maps of document collections.* Neurocomputing, 1998. **21**: p. 101-117.
12. Amarasiri, R., et al. *Enhancing Clustering Performance of Feature Maps Using Randomness*. in *Workshop on Self Organizing Maps (WSOM) 2005*. 2005. Paris, France.
13. Amarasiri, R., et al. *HDGSOMr: A High Dimensional Growing Self Organizing Map Using Randomness for Efficient Web and Text Mining*. in *IEEE/ACM/WIC Conference on Web Intelligence (WI) 2005*. 2005. Paris, France.
14. Wikipedia, *Randomness*. 2005.
15. Holland, J., *Genetic algorithms and the optimal allocations of trials.* SIAM Journal of Computing, 1973. **2**(2): p. 88-105.
16. Kirkpatrick, S., C.D. Gelatt(Jr), and M.P. Vecchi, *Optimization by Simulated Annealing.* Science, 1983. **220**(4598): p. 671-680.
17. Amarasiri, R., J. Ceddia, and D. Alahakoon. *Exploratory Data Mining Lead by Text Mining Using a Novel High Dimensional Clustering Algorithm*. in *International Conference on Machine Learning and Applications*. 2005. LA, USA: IEEE.

Zoomed Clusters

Jean-Louis Lassez, Tayfun Karadeniz, and Srinivas Mukkamala*

Department of Computer Science, Coastal Carolina University, Conway, SC 29528
* Institute for Complex Additive Systems and Analysis, Socorro, NM 87801
{Jlassez, tkarade}@coastal.edu, srinivas@cs.nmt.edu

Abstract. We use techniques from Kleinberg's Hubs and Authorities and kernel functions as in Support Vector Machines to define a new form of clustering. The increase in the degree of non linearity of the kernels leads to an increase in the granularity of the data space and to a natural evolution of clusters into subclusters. The algorithm proposed to construct zoomed clusters has been designed to run on very large data sets as found in web directories and bioinformatics.

1 Preliminaries

We consider the standard model word/document where a document is represented as a vector in $\Re^n$ whose coefficients represent a measure of the frequency of occurrence of the words in a dictionary of size n [1],[2]. Consequently we assume here that the vectors are of length 1 and in the positive quadrant. This restriction is made because of the applications we have in mind, but it can be removed or overcome easily if needed. We further assume that we have m documents, and that we use the dot product of vectors, that is the cosine of the angle formed by two vectors, to define a notion of similarity between vectors, 1 if the two vectors are identical, 0 if they are orthogonal.

A standard technique in the query process of search engines is to assume that two vectors close to each other represent two semantically related documents. In a first instance, one can use the similarity to the centroid of a set of related documents to determine if this set should be retrieved by a given document (the query). This will work well when the set of documents is tightly and evenly distributed around its centroid. So if a new document is found to be close to the centroid, it should also be found to be related to the whole set. Consider the following analogy: the sphere represents the earth, to a vector we associate its *elevation*, that is the measure of similarity to the centroid, the cosine of the angle formed by the two vectors. The set of documents then defines a simple volcano like conical shape. A threshold used to determine if a document should be retrieved defines a circular contour line above which are the documents to be retrieved. The elevation determines the degree of relevance. It seems clear that a more flexible model should allow contour lines ellipsoidal instead of circular, or even two disconnected circular contour lines. That would be the case if the data naturally formed two subclusters instead of one as assumed. Pursuing the analogy we would like to "zoom" on the volcano and refine the initially crude structure into a more complex one, realizing that a peak may in fact

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 824 – 830, 2006.

consist of several subpeaks. One could implement this idea using the k-means algorithm or its derivatives and using a notion of distance to the mean to represent elevations. However we will do it differently, in a way that is (arguably) more natural, but will also avoid two main pitfalls:

One is the uncertainty concerning the quality of the output (several are possible depending on the starting conditions). The other is the computational cost.

In a first step we consider the problem of outliers.

Let $\{D_i\}$ be a set of m documents written out of a dictionary of n words, the centroid C_1 of that set is therefore:

$$C_1 = 1/m \sum_{i=1}^{i=m} D_i$$

Note that since we are working with vectors of length 1, we will assume that all computed vectors are appropriately normalized by the function N . Now we want to “recenter” the centroid by penalizing outliers. Recursively, the vectors are weighted by their proximity to the current centroid:

$$C_n = 1/m \sum_{i=1}^{i=m} \langle D_i , N(C_{n-1}) \rangle D_i \tag{1}$$

We iterate the process and show that it converges:

Let M be the matrix whose rows are the vectors D_i, and whose columns W_i represent word frequencies,

Proposition

$C_n = 1/m \sum_{i=1}^{i=m} \langle D_i , N(C_{n-1}) \rangle D_i$ converges for n → ∞ towards the principal eigenvector $C\infty$ of the matrix M^TM, called the *iterated centroid.*

Proof

A little computation shows that $C_n = M^TMC_{n-1}$ up to a scalar. One first represents MC_{n-1} as a vector of vector products between the D_i’s and C_{n-1} and then shows that the multiplication by M^T can be written as the appropriate linear combination of the D_i’s.

We are then in a situation similar to [8]. The matrix M^TM is symmetric positive, by a variant of Perron Frobenius we know that it has a principal eigenvector which can be computed iteratively by successive powers of M^TM applied to any vector not orthogonal to the principal eigenvector.

Consider the hyperplane whose normal vector is the principal eigenvector, we can always find a vector in the first quadrant which is not in that hyperplane, and therefore not orthogonal to the eigenvector. As the matrix has only non negative entries, all the vectors generated by the matrix products will be in the first quadrant, including the principal eigenvector. We can assume that C_1 has non zero coefficients (otherwise all the D_i’s would also have one in the same position and we could reduce all the vectors by one dimension). C_1 having no coefficient equal to zero will not be orthogonal to any vector in the first quadrant, including the principal eigenvector.

We have therefore two methods for computing the iterated centroid, the iteration [1], and the matrix products.

We know from [8] that the computation by repeated matrix multiplication is very efficient even for the gigantic matrices that are found in search engines applications. We also know that we are less likely to have serious problems with round off errors because essentially all vectors will ultimately converge to the same limit. The iterated centroid can also of course be computed by the previously described process.

In this set up we have a variant of technique used in search engines to retrieve a single cluster of documents, those that are sufficiently close to a centroid [1]. It fits the single volcano analogy. We will now introduce the techniques that will allow us to "zoom" and reveal subclusters.

2 Mapping to Higher Dimensions

In the previous model, if the data in fact consists of two clearly separated subclusters, an isolated new vector in between these two subclusters will be closer to the centroid than the other vectors, and consequently it will be more eagerly retrieved than the initial vectors, even though it may be either an outlier or may have no relevance. Entering the query "simplex" in a standard search engine, returned over 300 first sites devoted to venereal diseases, before getting sites related to the algorithm. In this previous model, the documents with a highest rating would be those "in the middle" that is, for instance, documents applying operation research techniques to the spread of venereal diseases. But even though the user might find the documents relevant, it is unlikely that she/he would like those returned first. Now entering the word "bat" as a query would lead to the retrieval of documents on baseball and documents on nocturnal flying mammals. These two subclusters have little in common, but an hypothetical document on Batman playing baseball would receive the highest evaluation, while its relevance is clearly arguable. So there is indeed a need to separate into subclusters and appropriately assess the relevance of each document. And of course when we deal with enormous amounts of data, we do not know a priori how many times we should separate subclusters into further subclusters. The analogy with zooming on a mountain range which reveals increasingly separate peaks is one that seems to reflect our clustering needs most appropriately.

Once we have found a cluster of related documents we can use machine learning techniques to determine if a new document is relevant to these documents. Among the various techniques, Support Vector Machines have been particularly successful see, for instance to filter spam [5,6,10]. But a key aspect of SVM's is the generation of non linear surfaces to separate complex configurations of data points. We will use the same motivations as those used in Support Vector Machines theory and the same techniques of mapping to higher dimensions in order to achieve greater clustering flexibility while preserving computational efficiency. The reader is referred to [3, 4, 7] for a more thorough presentation of this issue and applications of SVM's. Let us just summarize here a few necessary basic facts.

Mapping in a higher dimension while preserving computational efficiency is achieved by functions Φfrom the initial space into a higher dimensional space such that there exists a so-called *kernel function K* with the following property:

$$K(D_i, D_j) = \langle \Phi(D_i), \Phi(D_j) \rangle$$

An example of such functions is:

$$\Phi r1,r2,\ldots,rd x) = \sqrt{p!/r_1!\,r_2!\ldots r_d!}\, x_1^{r1}\, x_2^{r2} \ldots\ldots x_d^{rd}$$

$$\text{with } k(x,y) = (<x,y>)^p = <\Phi(X),\Phi(Y)>$$

$$\sum_{i=1}^{i=d} r_i = p$$

What is quite significant here is that we can compute the vector products in the high dimensional space by simply applying a function to the initial vectors.

In other words even though the dimension of the higher dimensional space may be of exponential size, we can still compute efficiently vector products.

In fact, for mere computational purposes we do not even need to know the function □once we know its associated kernel $\tilde{K}$.

For instance the following kernel corresponds to a mapping into a space of infinite dimension:

$$K(x, y) = \tilde{e}^{//x-y//^2 2\sigma^{22}}$$

It is known as the Gaussian Radial Basis Function. A brilliant achievement of SVM's is that they can compute non linear separating surfaces for essentially the same cost as computing a linear separation. This is done by finding a formulation (Wolfe's dual formulation of a quadratic optimization problem) where the data appears only as vector products. By using the function *K* we are able to eliminate all occurrences of the function $\tilde{□}$ Indeed one can then simply replace <x,y> by *K*(x,y) and we have transformed a linear separation problem into a non linear one.

We remark here that the various kernels used in SVM's [3, 4] correspond to mappings Φ that map a sphere into a sphere, because *K(x,x)* is a constant for vectors *x* of norm 1. Therefore we can map our data in a space of higher dimension with a function Φthat has an associated kernel and, at least in principle, compute an iterated centroid in that space.

Unfortunately the two methods proposed to compute the iterated centroid do not allow us to use kernels. The iteration (1) becomes after the mapping by Φ:

$$F_n = 1/m \sum_{i=1}^{i=m} (< \Phi D_i), N(F_{n-1})>)\, \Phi D_i) \qquad (2)$$

And because F_{n-1} is in the higher dimension space, we cannot easily replace Φby *K.*

The matrix M has now rows made of the vectors $\Phi D_i)$, which are in a space of unmanageable dimension (eventually infinite), so we cannot compute practically iteratively the powers of the matrix M^TM. In other words in these approaches we cannot eliminate the computational use of the function Φ. We will solve this problem by allowing Φto remain, but only in a purely symbolic manner.

Instead of computing F_n at each step, we will only compute the coefficients of $F_n$so the iterated centroid will be computed as a linear combination ofΦD_i)'s. From

$$F_n= 1/m \sum_{i=1}^{i=m} (< \Phi \mathrm{D_i}), N(F_{n-1})>)\ \Phi D_i)$$

and up to scaling, we can see easily that the coefficient Φ_i^n of ΦD_i) at iteration n can be computed from the coefficients Φ_j^{n-1} from the previous iteration. Indeed:

$$\Phi_i^n =< \Phi \mathrm{D_i}), F_{n-1}> =< \Phi \mathrm{D_i}), \sum_{j=1}^{j=m} (< \Phi \mathrm{D_j}), F_{n-2}>) \Phi D_j)>) = < \Phi \mathrm{D_i}), \sum_{j=1}^{j=m} \Phi_j^{n-1} . \Phi D_j)>$$

we have removed the scaling for sake of notational clarity, it is not a problem as all Φ's are uniformly affected

Finally we have :

$$. \Phi_i^n = \sum_{j=1}^{j=m} \Phi_j^{n-1} k(D_i, D_j) \qquad (3)$$

We choose $\alpha_i^1 = < \Phi(\mathrm{D_i}), \mathrm{F_0}>$ with $\mathrm{F_0} = \Phi(\mathcal{N}(\sum_{i=1}^{i=m} \mathrm{D_i}))$

That is :

$$\alpha_i^1 = \mathrm{K}(\mathrm{D_i}, \mathcal{N}(\sum_{i=1}^{i=m} \mathrm{D_i}))$$

Because the centroid of the $\mathrm{D_i}$'s has no zero coefficient and because of the properties of the Φ's we consider, F_0 also has no zero coefficients.

We are now in a position to apply the proposition in the preliminaries to the present situation in a higher dimensional space, and we conclude that the iterated centroid is obtained as:

Theorem

$$F_\infty = \sum_{i=1}^{i=m} \alpha_i^* \Phi(\mathrm{D_i})$$

where

$$\alpha_i^* = lim \{\alpha_i^n = (1/m) \sum_{j=1}^{j=m} \alpha_j^{n-1} k(D_i, D_j)\}\ for\ n \longrightarrow \infty$$

and the elevations Φ_i^* are, up to scaling, the coordinates of the principal eigenvector of the matrix $[\mathrm{K}(\mathrm{D_i}, \mathrm{D_j})]$.

Now we can compute the elevation El of a vector D using only the function K.

$$\mathrm{El(D)} = \sum_{i=1}^{i=m} \Phi_i^* < \Phi \mathrm{D_i}), \Phi \mathrm{D})> = \sum_{i=1}^{i=m} \Phi_i^* K \mathrm{D_i}, \mathrm{D})$$

Which gives us a non linear evaluation of the likelihood of D to belong to the same group as the D_i's. Contour lines are non linear and may even be disconnected, revealing the presence of multiple peaks. We can split the data into several clusters as we now show.

3 Zoomed Clusters

If we extend the tip of the vectors by a quantity proportional to their elevations, that is by a number that reflects how close a vector is from the given vectors, we obtain a picture similar to mountain ranges. A linear kernel gives us a volcano like symmetrical conical peak, that is we have one cluster, and the elevation of a point tells us about how much it belongs to this cluster. With increasing non linearity we see several peaks appear, that is the points are now grouped into several clusters. Elevation by itself does not tell us which peak, or cluster, a point belongs to. For a fixed value of the elevation function we have contour lines, a point at a given elevation may be at the lower level of a high peak, or at the summit of a low peak. So the concept on which the clustering algorithm is based is that a point in a mountain range is associated to the *"highest peak in its vicinity"*. As a consequence the algorithm aims at telling us which cluster a point belongs to, but also how strongly it does belong to this cluster as well as how strongly it is related to the other clusters.

3.1 The Zoomed Cluster Algorithm

As input we have a set $\{D_i\}$ of documents, a kernel K, and an elevation function

$$\mathrm{El(D)}=\sum_{i=1}^{i=m} \Phi_i^{*} \mathrm{K}(D_i, D).$$

We construct a directed graph Γ whose nodes are the documents, from which we will derive clusters as being specific subgraphs.

We have an edge (edges) from document D_j to the document(s) $\{D_k\}$ which is (are) the highest in D_j's vicinity, that is the document(s) maximizing the product $\alpha_i{}^{*}<D_i, D_j>^{s}$. Where s is a parameter which can be used to smoothe the notion of vicinity when > 1, or sharpen it when < 1. A local maximum is a document which is the highest in its vicinity, we will call it a *peak.*

A cluster is defined as the set of documents from which we can reach a given peak by following edges. Of course we can use any kernel instead of $<D_i, D_j>^{s}$, the default being the same kernel as for the elevation function. For implementation purposes one can remark that the document with the highest elevation is a peak and therefore defines a cluster. The second highest document can either be part of this first cluster or it forms one itself, and so on. We see that the worst case will be achieved when all documents are peaks, so scattered that each forms a cluster by itself.

➢ Apart from the exceptional cases where there are more than one highest peak in the vicinity of a document, we will construct clusters with an empty intersection. To obtain clusters with non empty intersection we can simply modify the algorithm by creating links to all documents that are higher in the vicinity of the document under consideration, and not just to the highest. Following the mountain analogy, a cluster will be a set of paths leading to a peak, and several peaks may share subpaths.

- Implied clusters: Let D be a new and arbitrary document, and D_i be the highest document in D's vicinity, among the set of given documents. Add D to the clusters that D_i belongs to. The resulting clusters are called *implied clusters*. This notion can used for unsupervised classification purposes. Alternatively when we have data that we know does not belong a particular cluster , one can use the elevation function with a estimated threshold to act as a supervised classifier.

4 Summary and Future Work

We have presented a novel form of clustering which is very easy to implement, computationally efficient, and most importantly whose "zooming" capabilities seem more flexible and useful than having to guess an a priori number of clusters. Its real value however will be ascertained through experimentations with the large amounts of data as found on the web and in genomics. Some of these experimentations are near conclusion and their results are to be reported.

References

1. Berry,M.W., Dumais,S.T. and O'Brien,G.W. (1995) Using Linear Algebra For Intelligent Information Retrieval. SIAM Review, 37(4), 573-595.
2. Burges,C. (1998) A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2(2), 121-167.
3. Cristianini,N. and Shawe-Taylor,J. (2000) An Introduction to Support Vector Machines. Cambridge University Press. Cambridge.
4. Drucker,H., Wu,D. and Vapnick,V.N. (1999) Support vector machines for spam categorization. IEEE Transactions on Neural Networks, 10(5), 1048-1054.
5. Dumais,S.T., Platt,J., Heckerman,D. and Sahami,M. (1998). Inductive Learning Algorithms and Representations for Text Categorization. Proceedings of ACM-CIKM98, 148-155.
6. Guyon,I. http://www.clopinet.com/isabelle.
7. Kleinberg J., Authoritative Sources in a Hyperlinked Environment Journal of the ACM 46 (1999)
8. Thorsten J. http://ais.gmd.de/~thorsten/svm_light.
9. Thorsten J. (1998) Text Categorization with Support Vector Machine Learning With Many Relevant Features. European Conference on Machine Learning.
10. Vapnik,V.N. (1998) Statistical Learning Theory. John Wiley & Sons.

Predicting Chaotic Time Series by Boosted Recurrent Neural Networks

Mohammad Assaad, Romuald Boné, and Hubert Cardot

Université François Rabelais de Tours,
Laboratoire d'Informatique,
64, avenue Jean Portalis, 37200 Tours, France
{mohammad.assaad, romuald.bone,
hubert.cardot}@univ-tours.fr

Abstract. This paper discusses the use of a recent boosting algorithm for recurrent neural networks as a tool to model nonlinear dynamical systems. It combines a large number of RNNs, each of which is generated by training on a different set of examples. This algorithm is based on the boosting algorithm where difficult examples are concentrated on during the learning process. However, unlike the original algorithm, all examples available are taken into account. The ability of the method to internally encode useful information on the underlying process is illustrated by several experiments on well known chaotic processes. Our model is able to find an appropriate internal representation of the underlying process from the observation of a subset of the states variables. We obtain improved prediction performances.

Keywords: Boosting, Time series forecasting, Recurrent neural networks, Chaotic time series.

1 Introduction

Predicting the future evolution of dynamical systems has been a main goal of scientific modeling for centuries. Chaotic phenomena frequently appear in economics, meteorology, chemical processes, biology, hydrodynamics and many other situations. In order to predict and/or control the underlying systems, mathematical models are generally investigated analyzing the evolution properties of the corresponding equations of motion, and integrating them in order to predict the future state of the system. But in the absence of prior knowledge concerning the problem to solve, one must build models of time series out of available data. A common approach to modeling is to consider a fixed number of the past values of one or several time series (a time window) and look for a function which provides the next value of the target series or the class membership of the input sequence.

Multilayer perceptrons or MLPs are well adapted to this approach. Universal approximation results show that very general nonlinear autoregressive (NAR) functions can be obtained. But the use of an MLP for time series prediction has inherent limitations, since one cannot find an appropriate finite NAR model for every

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 831 – 840, 2006.

dynamical system. Also, the size of the time window is difficult to choose; an optimal value may depend on the context.

Recurrent neural networks (RNNs) possess an internal memory and do no longer need a time window to take into account the past values of the time series. RNNs are computationally more powerful than feed-forward networks [1], and valuable approximation results were obtained for dynamical systems [2]. To improve upon the obtained performance, we can adapt general procedures that were found to enhance the accuracy of various basic models. One such procedure is known under the name of boosting and was introduced in [3]. The gain a learner bring with respect to random guessing is boosted by the sequential construction of several such learners, progressively focused on difficult examples of the original training set.

This paper deals with boosting of RNNs for improving forecasting of chaotic time series. In section 2, we review existing approaches for nonlinear modeling and the corresponding neural architectures before turning in section 3 to a presentation of our algorithm. The experimental results obtained on different benchmarks, showing an improvement in performance, are described in section 4.

2 Nonlinear Modeling Using RNN

The problem of designing a neural model of an unknown process based on observed data, without any physical insight in the underlying dynamics, has attracted much attention during the past years [4] [5] [6]. Most of the literature is concerned with black-box modeling usually performed using input-output models. This paragraph provides an understanding of their limitations. We show that state-space models such as RNN constitute a broader class of nonlinear dynamical models capable of remedying these limitations.

2.1 The NARMA Approach

We briefly review the cornerstone of black-box modeling in order to highlight the role of neural networks in approximating the optimal predictors of nonlinear dynamical systems [7]. Before taking a closer look at the existing approaches we must introduce some notation. Consider $x(t)$, for $0 \le t \le T$, the time series data one can employ for building a model. Given $\{x(t-1), x(t-2), \ldots, x(t-n), \ldots x(1)\}$ one is looking for a good estimate $\hat{x}(t)$ of $x(t)$.

The most common approach in dealing with a prediction problem consists in using a fixed number *p* of past values (a fixed-length time window sliding over the time series) when building the prediction:

$$\hat{x}(t) = f(x(t-1), x(t-2), \ldots, x(t-p)) \qquad (1)$$

where f is an unknown function. If f is a linear function, we obtain an autoregressive model. Most of the current work relies on a result in [8] showing that under several assumptions (among which the absence of noise) it is possible to obtain a perfect estimate of $x(t)$ according to (1) if $p \ge 2d + 1$, where d is the dimension of the stationary attractor generating the time series.

Another model of the class which appeared to offer plausible description of a wide range of different types of time series is the nonlinear extension of the autoregressive moving average model termed NARMA(*p*,*q*) in the literature:

$$x(t) = f\left(x(t-1), x(t-2), \ldots x(t-p), e(t-1), e(t-2), \ldots, e(t-q)\right) + e(t) \tag{2}$$

Suppose that the NARMA is invertible; there exists a function *g* such that

$$x(t) = g\left(x(t-1), x(t-2), \ldots\right) + e(t) \tag{3}$$

then given the infinite past observations, one can in principle use the above equation to compute the *e(t-j)* in (2) as a function of the past observations *x(t-*1*)*, *x(t-*2*)*, ... such that (2) becomes

$$\hat{x}(t) = f\left(x(t-1), \ldots x(t-p), e(t-1), \ldots, e(t-q)\right) \tag{4}$$

where the *e(t-j)* are specified by $x(t) - g\left(x(t-1), x(t-2), \ldots\right)$. Since in practice one has only access to a finite observation record, *e(t-j)* cannot be computed exactly; it seems reasonable, as for the linear ARMA process, to approximate (4) by:

$$\hat{x}(t) = f\left(x(t-1), \ldots x(t-p), \hat{e}(t-1), \ldots, \hat{e}(t-q)\right) \tag{5}$$

where $\hat{e}(t-i) = x(t-i) - \hat{x}(t-i)$. The optimal predictor is thus given by (5) provided that the effect of arbitrary initial conditions will die away depending on the unknown function *f*. Such NARMA models have an advantage over NAR models in much the same way that linear ARMA models have advantages over AR for some types of series. However, *p* and *q* values must be chosen carefully. Moreover, in contradiction to the linear case, there is no simple equivalence between nonlinear input-ouput and state-space models. Therefore, a natural step is to resort to state-space models.

2.2 The State Space Approach

Whereas it is always possible to rewrite a nonlinear input-output model in a state-space representation, conversely, an input-output model equivalent to a given state-space model might not exist [9], and if it does, it is surely of higher order [4]. Consider the following deterministic state-space model:

$$\begin{cases} \mathbf{x}(t+1) = f\left(\mathbf{x}(t), u(t)\right) \\ y(t) = g\left(\mathbf{x}(t)\right) \end{cases} \tag{6}$$

where *u(t)* is a scalar external input, *y(t)* is the scalar output, and $\mathbf{x}(t)$ is the *n*-state vector of the model at time *t*. Under general conditions on the observability of the system, an equivalent input-output number of past model does exist, and is given by:

$$y(t) = h\left(y(t-1), \ldots, y(t-r), u(t-1), \ldots, u(t-r)\right) \tag{7}$$

with $n \leq r \leq 2n+1$ [4]. Therefore, state-space models are likely to have lower order and require a smaller number of past inputs, and hopefully a smaller number of parameters. State-space models can be used as black-box models with additional outputs dedicated to modeling the state variables of the process. This gives to the neural predictor more flexibility while still taking advantage of the feed-forward structure. Consider the stochastic state-space model given by:

$$\begin{cases} \mathbf{x}(t+1) = f\left(\mathbf{x}(t), u(t), \boldsymbol{\varepsilon}_1(t)\right) \\ y(t) = g\left(\mathbf{x}(t), \boldsymbol{\varepsilon}_2(t)\right) \end{cases} \tag{8}$$

where $\boldsymbol{\varepsilon}_1(t)$and $\boldsymbol{\varepsilon}_2(t)$ are sequences of zero mean i.i.d. random independent vectors. If f and g are known, the extended Kalman predictor gives a sub-optimal solution using a linearization of the model around the current state estimate. In practice however, these functions are partially or completely unknown, there is thus no reason to stick to this structure. Such time-invariant associated predictor can be put in general form:

$$\begin{cases} \hat{\mathbf{x}}(t+1) = N_1\left(\hat{\mathbf{x}}(t), y(t)\right) \\ \hat{y}(t+1) = N_2\left(\hat{\mathbf{x}}(t+1)\right) \end{cases} \tag{9}$$

where N_j ; j = 1; 2 may be implemented by feed-forward networks for instance. This state-space based model can be modeled by two cascaded MLPs. However, a MLP in which the output is fed back to the input is a special case of the somewhat more general class of fully interconnected networks [10] [11]. For such recurrent networks, there is no need to present the $\hat{e}(t-j)$ and the state variables $\mathbf{x}_{t-j}$, j >1, to the input since specific non trainable recurrent links from a hidden layer towards itself results in the same functional input-output mapping.

3 Boosting Recurrent Neural Networks

To improve the obtained results, we may use a combination of models to obtain a more precise estimate than the one obtained by a single model. In the boosting algorithm, the possible small gain a “weak” model can bring compared to random estimate is boosted by the sequential construction of several such models, which concentrate progressively on the difficult examples of the original training set. The boosting [3] [12] [13] works by sequentially applying a classification algorithm to re-weighted versions of the training data, and then taking a weighted majority vote of the sequence of classifiers thus produced. Freund and Schapire in [12] outline their ideas for applying the Adaboost algorithm to regression problems; they presented the Adaboost.R algorithm that attacks the regression problem by reducing it to a classification problem.

Recently, a new approach to regressor boosting as residual-fitting was developed [14] [15]. Instead of being trained on a different sample of the same training set, as in previous boosting algorithms, a regressor is trained on a new training set having different target values (e.g. the residual error of the sum of the previous regressors). Before presenting our algorithm, let us mention the few existing applications of boosting to time series modelling. In [16] a boosting method belonging to the family of boosting algorithms presented in [3] is applied to the classification of phonemes.

The learners employed are RNNs, and the authors are the first to notice the implications the internal memory of the RNNs has on the boosting algorithm. A similar type of boosting algorithm is used in [17] for the prediction of a benchmark time series, but with MLPs as regressors.

Table 1. The boosting algorithm proposed for regression with recurrent neural networks

```
1. Initialize the weights for the examples: $D_1(q) = 1/Q$, and $Q$, the number of training examples. Put the iteration counter at 0: $n = 0$
2. Iterate
    (a) increment $n$. Learn with BPTT [18] a RNN $h_n$ by using the entire training set and by weighting the squared error computed for example $q$ with $D_n(q)$, the weight of example $q$ for the iteration $n$;
    (b) update the weights of the examples:
        (i) compute $L_n(q)$ for every $q = 1, \cdots, Q$ according to the loss function :
            $L_n^{linear}(q) = \left|y_q^{(n)}(x_q) - y_q\right| / S_n$, $L_n^{quadratic}(q) = \left|y_q^{(n)}(x_q) - y_q\right|^2 / S_n^2$
            $L_n^{exponential}(q) = 1 - \exp\left(-\left|y_q^{(n)}(x_q) - y_q\right| / S_n\right)$, with
            $S_n = \sup_q \left|y_q^{(n)}(x_q) - y_q\right|$ ;
        (ii) compute $\varepsilon_n = \sum_{q=1}^{Q} D_n(q) L_n(q)$ and $\alpha_n = (1 - \varepsilon_n)/\varepsilon_n$ ;
        (iii) the weights of the examples become ($Z_n$ is a normalizing constant)
            $D_{n+1}(q) = \frac{1 + k \cdot p_{n+1}(q)}{Q + k}$ with $p_{n+1}(q) = \frac{D_n(q)\alpha_n^{(L_n(q)-1)}}{Z_n}$ until $\varepsilon_n < 0.5$.
3. Combine RNNs by using the weighted median.
```

Our new boosting algorithm should comply with the restrictions imposed by the general context of application. In our case, it must be able to work well when a limited amount of data is available and accept RNNs as regressors. We followed the generic algorithm of [19]. We had to decide which loss function to use for the regressors, how to update the distribution on the training set and how to combine the resulting regressors. Our updates are based on the suggestion in [20], but we apply a linear transformation to the weights before employing them (see the definition of $D_{n+1}(q)$ in the Table 1) in order to prevent the RNNs from simply ignoring the easier examples for problems similar to the sunspots dataset. Then, instead of sampling with replacement according to the updated distribution, we prefer to weight the error computed for each example (thus using all the data points) at the output of the RNN with the distribution value corresponding to the example.

4 Experiments

In this section, we report on extensive investigations of the performance of the boosted RNNs as an anticipative model for the behavior of the well known time series generated by chaotic processes. Our main goals were to assess the predictive ability of our method against other forecasting techniques, especially when no past information $x(t\text{-}1)$, $x(t\text{-}2)$, … is directly supplied to the network for the prediction of $x(t+1)$, and to force the dynamical network to encode as much information of the past as possible to infer some form of embedding dimension regarding the underlying process. In a previous paper [21], we gave some basic results on our algorithm.

We will now come back to a more detailed study, focusing on chaotic time series, providing results after 5 trial runs for each configuration: (linear, squared or exponential loss functions; value of the parameter k). The error criterion used was the normalized mean squared error (NMSE), a standard measure of fit, which is the ratio between the MSE and the variance. A NMSE value of 1 corresponds to the prediction, for all time steps, of the mean of a time series. To obtain the best mean results in following tables, we take the normalised mean results of the 5 trials of each set of parameters, and then we choose the best results.

The employed architectures had a single input neuron, a single linear output neuron, a bias unit and a fully recurrent hidden layer composed of neurons with tanh activation functions. The numbers of neurons correspond to the best results obtained by BPTT without boosting. We set the maximal number n of RNNs at 50 for each experiment and for each one the weights in $[-0.3, 0.3]$ are randomly initialized. We compared the results given by our algorithm to other results in the literature (see [11] for more details).

4.1 Mackey-Glass Datasets

The Mackey-Glass time series [22], well-known for the evaluation of forecasting methods [23], are generated by the following non-linear differential equation:

$$\frac{dx(t)}{dt} = -0.1x(t) + \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} \tag{10}$$

Depending on the value of τ, the time-series generated can asymptotically converge to a fixed point, to a periodic behavior or to deterministic chaos for $\tau > 16{,}8$. The results in the literature usually concern $\tau = 17$ (known as MG17) and $\tau = 30$ (MG30). The data generated with $x(t) = 0.9$ for $0 \le t \le \tau$ is then sampled with a period of 6, according to the common practice (see e.g. [24]). We use the first 500 values for the learning set and the next 100 values for the test set. We tested RNNs having 7 neurons in the hidden layer. Tables 2 and 3 show the strong improvements obtained on several models applied to this benchmark (see [11] for a presentation). Our boosting algorithm significantly improves upon the mean results and is close to the best results reported in the literature for the two datasets. For most of the simulations, 50 networks have been obtained, which is the maximal number of

Table 2. Best results (NMSE*10^3)

Model	MG17	MG30
Linear	269	324
Polynomial	11.2	39.8
RBF	10.7	25.1
MLP	10	31.6
FIR MLP	4.9	16.2
TDFFN	0.8	_
DRNN	4.7	7.6
RNN/BPTT	0.23	0.89
EBPTT	0.13	0.05
CBPTT	0.14	0.73
Boosting (linear, 150)	0.13	0.45
Boosting (squared, 100)	0.15	0.41

Table 3. Best mean results (NMSE*10^3)

Model	MG17	MG30
RNN/BPTT	4.4	13
EBPTT	0.62	1.84
CBPTT	1.6	2.5
Boosting (squared, 100)	0.16	0.45
Boosting (squared, 200)	0.18	0.45

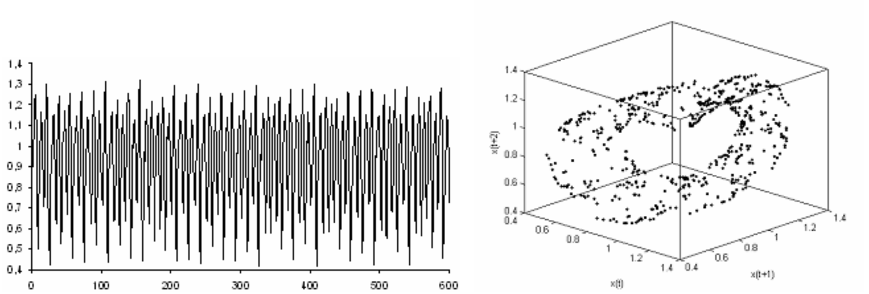


Fig. 1. The Mackey-Glass time series and attractor for $\tau = 17$ (learning and test set)

networks that has been allowed. The MG17 attractor is reconstructed without default (Fig. 1). Several additional models applied to this last time series can be found in [25], some of them with better results than the mentioned models but obtained from a different dataset (number of data, sampling, …).

4.2 Hénon Dataset

The Hénon attractor is defined by the bi-dimensional system:

$$\begin{cases} x(t) = x(t) = 1{,}0 - 1{,}4x^2(t-1) + 0{,}3x(t-2) \\ y(t) = x(t-1) \end{cases} \tag{11}$$

The phase plot, shown in Fig. 2, reveals a remarkable structure called a strange attractor of dimension D = 1.26 [26]. It is the region in which the trajectory of states is confined to. Increased magnification of the attractor would reveal ever finer detail in a fractal like geometry. A RNN with 5 hidden neurons was trained on a learning set of 5000 values ([10], [27]). Validation and test set contain 1500 values. As it can be seen in Fig. 2 and tables 4 and 5, the boosted RNN prediction are remarkably accurate. The average numbers of networks generated by boosting are 11, 10, and 26 respectively for the linear, squared and exponential loss functions.

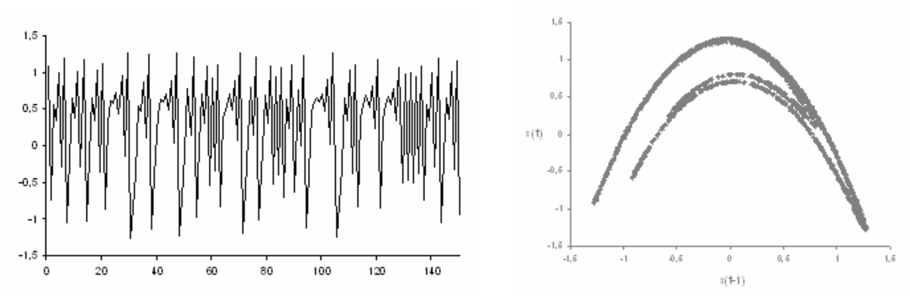


Fig. 2. The Hénon time series and attractor (test set)

Table 4. Best results (NMSE*10^3)

Model	Henon
Linear	874
FIR MLP	1.7
DRNN	1.2
BPTT	0.2
EBPTT	0.012
CBPTT	0.030
CBPTT si	0.062
CBPTT 20 it.	0.023
Boosting (linear, 20)	0.011
Boosting (squared, 150)	0.010
Boosting (exponential, 100)	0.015

Table 5. Best mean results (NMSE*10^5)

Model	Henon
BPTT	58.6
EBPTT	25.0
CBPTT	64.1
CBPTT si	39.4
CBPTT 20 it.	26.3
Boosting (linear, 100)	2.82
Boosting (squared, 150)	3.02
Boosting (exponential, 100)	2.26

4.3 Laser Dataset (Lorenz Equations)

These data (Fig. 3) were recorded from a Far-Infrared-Laser in a chaotic state and was one of the datasets employed in the Santa Fe Institute time series prediction and analysis competition in 1991. The data is a cross-cut through periodic to chaotic intensity pulsations of the laser. Chaotic pulsations more or less follow the theoretical Lorenz model.

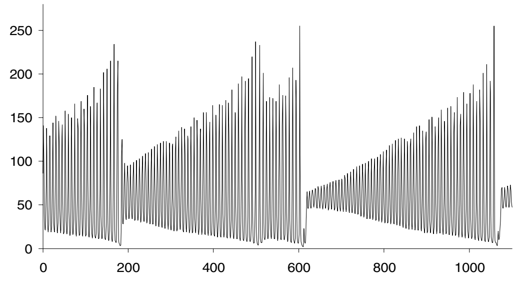


Fig. 3. The laser time series

Table 6. Best results (NMSE*10^3) on test set

Model	Laser
FIR MLP	23
BPTT	7.92
EBPTT	5.37
CBPTT	5.60
Boosting (linear, 20)	3.77
Boosting (squared, 5)	4.31

In accordance with the instructions given to the competitors back in 1991, we used the first 1000 values for the learning set and the next 100 values for the test set. We limited our experiments to single step predictions.

Table 6 compares the results obtained by our method to those from literature. One more time, we obtain improved results (table 6) with respectively 14 and 19 networks developed by our method.

5 Conclusion

In this paper, we have shown that boosted recurrent neural networks are valuable for modeling general nonlinear dynamical systems. A number of experiments on deterministic chaotic processes were carried out to confirm and to illustrate the ability of boosted RNN models to infer an internal representation of the nonlinear processes from the observation of a subset of the state variables. Our algorithm increases prediction performances and overcomes various results reported in the literature.

The evaluation on chaotic time series multi-step-ahead prediction problems is one of our further works on this algorithm.

References

1. Siegelmann, H.T., Horne, B.G., Giles, C.L.: Computational Capabilities of Recurrent NARX Neural Networks. IEEE Transactions on Systems, Man and Cybernetics 27 (1997) 209-214
2. Seidl, D.R., Lorenz, R.D.: A Structure by which a Recurrent Neural Network Can Approximate a Nonlinear Dynamic System. International Joint Conference on Neural Networks (1991) 709-714
3. Schapire, R.E.: The Strength of Weak Learnability. Machine Learning 5 (1990) 197-227
4. Levin, A.U., Narendra, K.S.: Control of Nonlinear Dynamical Systems Using Neural Networks. IEEE Transactions on Neural Networks 7 (1996) 30-42.
5. Oliveira, K.A., Vannucci, A., Silva, E.C.: Using Artificial Neural Networks to Forecast Chaotic Time Series. Physica A (2000) 393-399
6. Tronci, S., Giona, M., Baratti, R.: Reconstruction of Chaotic Time Series by Neural Models: a Case Study. Neurocomputing 55 (2003) 581-591
7. Connor, J.T., Martin, R.D., Atlas, L.E.: Recurrent Neural Networks and Robust Time Series Prediction. IEEE Transactions on Neural networks 5 (1994) 240-254
8. Takens, F.: Detecting Strange Attractors in Turbulence. In: Dynamical Systems and Turbulence. Springer-Verlag (1980) 366-381

9. Leontaritis, I.J., Billings, S.: Input-Output Parametric Models for Non-Linear Systems. International Journal of Control 41 (1985) 303-344
10. Aussem, A.: Dynamical Recurrent Neural Networks: Towards Prediction and Modelling of Dynamical Systems. Neurocomputing 28 (1999) 207-232
11. Boné, R., Crucianu, M., Asselin de Beauville, J.-P.: Learning Long-Term Dependencies by the Selective Addition of Time-Delayed Connections to Recurrent Neural Networks. NeuroComputing 48 (2002) 251-266
12. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences 55 (1997) 119-139
13. Ridgeway, G., Madigan, D., Richardson, T.: Boosting Methodology for Regression Problems. Artificial Intelligence and Statistics (1999) 152-161
14. Mason, L., Baxter, J., Bartlett, P.L., Frean, M.: Functional Gradient Techniques for Combining Hypotheses. In: Smola, A.J. *et al.* (eds.): Advances in Large Margin Classifiers. MIT Press (1999) 221-247
15. Duffy, N., Helmbold, D.: Boosting Methods for Regression. Machine Learning 47 (2002) 153-200
16. Cook, G.D., Robinson, A.J.: Boosting the Performance of Connectionist Large Vocabulary Speech Recognition. International Conference in Spoken Language Processing (1996) 1305-1308
17. Avnimelech, R., Intrator, N.: Boosting Regression Estimators. Neural Computation 11 (1999) 491-513
18. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Internal Representations by Error Propagation. In: Rumelhart, D.E., McClelland, J. (eds.): Parallel Distributed Processing: Explorations in the Microstructure of Cognition. MIT Press (1986) 318-362
19. Freund, Y.: Boosting a Weak Learning Algorithm by Majority. Workshop on Computational Learning Theory (1990) 202-216
20. Drucker, H.: Boosting Using Neural Nets. In: Sharkey, A. (ed.): Combining Artificial Neural Nets: Ensemble and Modular Learning. Springer (1999) 51-77
21. Boné, R., Assaad, M., Crucianu, M.: Boosting Recurrent Neural Networks for Time Series Prediction. International Conference on Artificial Neural Networks and Genetic Algorithms (2003) 18-22
22. Mackey, M., Glass, L.: Oscillations and Chaos in Physiological Control Systems. Science (1977) 197-287
23. Casdagli, M.: Nonlinear Prediction of Chaotic Time Series. Physica 35D (1989) 335-356
24. Back, A., Wan, E.A., Lawrence, S., Tsoi, A.C.: A Unifying View of some Training Algorithms for Multilayer Perceptrons with FIR Filter Synapses. Neural Networks for Signal Processing IV (1994) 146-154
25. Gers, F., Eck, D., Schmidhuber, J.: Applying LSTM to Time Series Predictable Through Time-Window Approaches. International Conference on Artificial Neural Networks (2001) 669-675
26. Ott, E.: Chaos in Dynamical Sytems. 1993: Cambridge University Press
27. Wan, E.A.: Time Series Prediction by Using a Connection Network with Internal Delay Lines. In: Weigend, A.S., Gershenfeld, N.A. (eds.): Time Series Prediction: Forecasting the Future and Understanding the Past. Addison-Wesley (1994) 195-217

Uncertainty in Mineral Prospectivity Prediction

Pawalai Kraipeerapun[1], Chun Che Fung[2], Warick Brown[3], Kok Wai Wong[1],
and Tamás Gedeon[4]

[1] School of Information Technology, Murdoch University, Australia
{p.kraipeerapun, k.wong}@murdoch.edu.au
[2] Centre for Enterprise Collaboration in Innovative Systems, Australia
l.fung@murdoch.edu.au
[3] Centre for Exploration Targeting, The University of Western Australia, Australia
wbrown@cyllene.uwa.edu.au
[4] Department of Computer Science, The Australian National University, Australia
tom@cs.anu.edu.au

Abstract. This paper presents an approach to the prediction of mineral prospectivity that provides an assessment of uncertainty. Two feed-forward backpropagation neural networks are used for the prediction. One network is used to predict degrees of favourability for deposit and another one is used to predict degrees of likelihood for barren, which is opposite to deposit. These two types of values are represented in the form of truth-membership and false-membership, respectively. Uncertainties of type error in the prediction of these two memberships are estimated using multidimensional interpolation. These two memberships and their uncertainties are combined to predict mineral deposit locations. The degree of uncertainty of type vagueness for each cell location is estimated and represented in the form of indeterminacy-membership value. The three memberships are then constituted into an interval neutrosophic set. Our approach improves classification performance compared to an existing technique applied only to the truth-membership value.

1 Introduction

The prediction of new mineral deposit location is a crucial task in mining industry. In recent years, Geographic Information System (GIS) and neural networks have been applied in many applications for mineral prospectivity prediction [1,2,3]. Several sources of data such as geology, geochemistry, and geophysics are involved in the prediction. Data collected from these sources always contains uncertainty. Hence, the predicted mineral deposit locations also contain some degrees of uncertainty. There are several types of uncertainty such as error, inaccuracy, imprecision, vagueness, and ambiguity [4,5]. This paper deals with two types of uncertainty, which are uncertainty of type error and uncertainty of type vagueness. Error can happen from several aspects such as measurement, data entry, processing, lacking of knowledge about data, or lacking of ability in measurement [5]. This study deals with error occurred in the process of prediction. Vagueness refers to boundaries that cannot be defined precisely [5]. In

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 841–849, 2006.

this study, the locations are known, but uncertain existence of favourability for deposit. Some locations have one hundred percent of favourability for deposits. Some locations have zero percent of favourability for mineral deposits. These cells are determined as non-deposit or barren cells. Most locations have degrees of favourability between these two extremes. For each location, we cannot predict the exact boundary between favourability for deposit and likelihood for barren. Vagueness or indeterminable information always occurs in the boundary zone.

This paper presents a method using GIS data and neural networks for predicting the degree of favourability for mineral deposit, degree of likelihood for barren, and degree of indeterminable information in the mineral prospectivity prediction. Instead of considering only uncertainty in the boundary between both degrees of favourability for deposit and barren, we also consider uncertainty of type error in the prediction of both degrees. A multidimensional interpolation method is used to estimate these errors. In order to represent the three degrees for each location, an interval neutrosophic set [6] is used to express them. The basic theory of an interval neutrosophic set is described in the next section.

The rest of this paper is organized as follows. Section 2 presents the basic theory of interval neutrosophic sets. Section 3 explains proposed methods for mineral prospectivity prediction and quantification of uncertainties using GIS data, neural networks, interval neutrosophic sets, and a multidimensional interpolation. Section 4 describes the GIS data set and the results of our experiments. Conclusions and future work are presented in Section 5.

2 Interval Neutrosophic Set

The membership of an element to the interval neutrosophic set is expressed by three values: t, i, and f. These values represent truth-membership, indeterminacy-membership, and false-membership, respectively. The three memberships are independent. In some special cases, they can be dependent. These memberships can be any real sub-unitary subsets and can represent imprecise, incomplete, inconsistent, and uncertain information [7]. In this paper, the three memberships are considered to be dependent. They are used to represent uncertainty information. This research follows the definition of interval neutrosophic sets that is defined in [7]. This definition is described below.

Let X be a space of points (objects). An interval neutrosophic set in X is defined as:

$$\begin{gathered} A = \{x(T_A(x), I_A(x), F_A(x)) | x \in X \wedge \\ T_A : X \longrightarrow [0,1] \wedge \\ I_A : X \longrightarrow [0,1] \wedge \\ F_A : X \longrightarrow [0,1]\} \end{gathered} \tag{1}$$

where

T_A is the truth-membership function,
I_A is the indeterminacy-membership function, and
F_A is the false-membership function.

3 Mineral Prospectivity Prediction and Quantification of Uncertainty

In this study, gridded map layers in a GIS database are used to predict mineral prospectivity. Fig.1 shows our proposed model that consists of GIS input layers, two neural networks, and a process of indeterminacy calculation. The output of this model is an interval neutrosophic set in which each cell in the output consists of three values: deposit output, indeterminacy output, and non-deposit output which are truth-membership, indeterminacy-membership, and false-membership values, respectively.

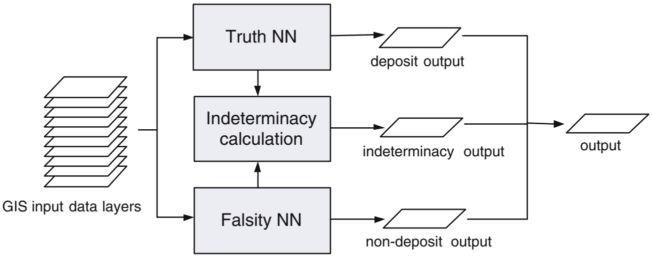


Fig. 1. Uncertainty model based on the integration of interval neutrosophic sets (INS) and neural networks (NN)

In the proposed model, the truth NN is a feed-forward backpropagation neural network. This network is trained to predict the degree of favourability for deposit, which is the truth-membership value. The falsity NN is also a feed-forward backpropagation neural network in which its architecture and all properties are the same as the architecture and all properties used for the truth NN. The only difference is that the falsity NN is trained to predict degree of likelihood for barren using the complement of target outputs used for training data in the truth NN. For example, if the target output used to train the truth neural network is 0.9, its complement is 0.1. Fig.2 shows our training model. It consists of two neural networks: truth NN and falsity NN. Errors produced from both neural networks will be used to estimate uncertainties in the prediction for the new input data.

Fig.3 shows uncertainty estimation in the prediction of truth-membership for the new data set or unknown data. The errors produced from the truth NN are plotted in the multidimensional feature space of the training input patterns. Thus, uncertainties of the new input patterns can be estimated using multidimensional interpolation. Estimated uncertainty in the barren prediction are also calculated in the same way as the estimated uncertainty for deposit. The errors produced from the falsity NN are plotted in the multidimensional feature space of the training input patterns. A multidimensional interpolation is then used to estimate uncertainty for the prediction of false-membership or degree of likelihood for barren. These two estimated uncertainties will be used in the dynamically

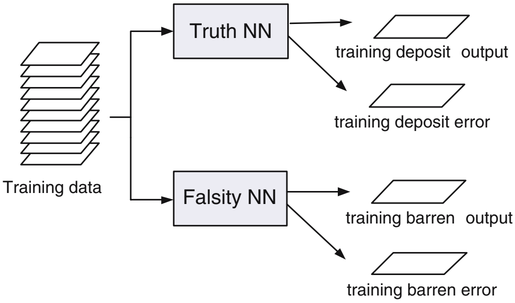


Fig. 2. Two neural networks used for training

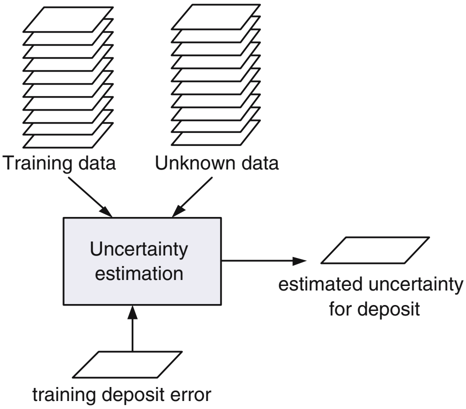


Fig. 3. Uncertainty estimation for mineral deposit prediction

weighted combination between truth-membership and false-membership for the binary classification later.

If the degree of favourability for deposit or truth-membership value is high then the degree of likelihood for barren or false-membership value should be low, and the other way around. For example, if the degree of favourability for deposit is 1 and the degree of likelihood for barren is 0, then the boundary between these two values is sharp and the uncertainty of type vagueness is 0. However, the values predicted from both neural networks are not necessary to have a sharp boundary. For instance, if both degrees predicted from the truth NN and the falsity NN for the same cell is equal, then this cell contains the highest uncertainty value, which is 1. Therefore, uncertainty in the boundary zone can be calculated as the difference between these two values. If the difference between these two values is high then the uncertainty is low. If the difference is low then the uncertainty is high. In this paper, uncertainty in the boundary between these two values is represented by the indeterminacy-membership value. Let C be an output GIS layer. $C = \{c_1, c_2, ..., c_n\}$ where c_i is a cell at location i. Let $T(c_i)$ be a truth-membership value at cell c_i. Let $I(c_i)$ be an indeterminacy-membership value at cell c_i. Let $F(c_i)$ be a false-membership value at cell c_i. For each cell, the indeterminacy membership value ($I(c_i)$) can be defined as follows:

$$I(c_i) = 1 - |T(c_i) - F(c_i)| \tag{2}$$

After three membership values are created for each cell, the next step is to classify the cell into either deposit or barren. Both truth-membership and false-membership are used in the classification. The estimated uncertainty of type error in the prediction of truth- and false-memberships are also integrated into the truth- and the false-membership values to support certainty of the classification. The less uncertainty in the prediction, the more certainty in the classification. Let $e_t(c_i)$ be an estimated uncertainty of type error in the prediction of the truth-membership at cell c_i. Let $e_f(c_i)$ be an estimated uncertainty of type error in the prediction of the false-membership at cell c_i. We determine the weights dynamically based on these estimated uncertainties. The weights for the truth- and false-membership values are calculated as the complement of the errors estimated for the truth- and false-membership, respectively. These weights are considered as the degrees of certainty in the prediction. In this paper, the certainty in the prediction of the false-membership value is considered to be equal to the certainty in the prediction of non-false-membership value, which is the complement of the false-membership value. Let $w_t(c_i)$ and $w_f(c_i)$ be the weights of the truth- and false-membership values, respectively. The output $O(c_i)$ of the dynamic combination among the truth-memberships, the false-memberships, and their uncertainties of type error can be calculated using equations below.

$$O(c_i) = (w_t(c_i) \times T(c_i)) + (w_f(c_i) \times (1 - F(c_i))) \tag{3}$$

$$w_t(c_i) = \frac{1 - e_t(c_i)}{(1 - e_t(c_i)) + (1 - e_f(c_i))} \tag{4}$$

$$w_f(c_i) = \frac{1 - e_f(c_i)}{(1 - e_t(c_i)) + (1 - e_f(c_i))} \tag{5}$$

In order to classify the cell into either deposit or barren, we compare the output to the threshold value. A range of threshold values are determined and compared to the output for each cell. The best threshold value that can produce the best accuracy in the classification will be selected for the mineral prospectivity prediction.

4 Experiments

4.1 GIS Data Set

The data set used in this study contains ten GIS layers in raster format. Each layer represents different variables which are collected and preprocessed from various sources such as geology, geochemistry, and geophysics in the Kalgoorlie region of Western Australia. An approximately 100 × 100 km area is divided into a grid of square cells of 100 m side. Each layer contains 1,254,000 cells. Each grid cell represents a single attribute value which is scaled to the range $[0, 1]$. For example, a cell in a layer representing the distance to the nearest fault contains a value of distance scaled to the range $[0, 1]$. Each single grid cell is classified into deposit or barren cell. The cells containing greater than 1,000 kg total contained gold are labeled as deposits. All other cells are classified as non-deposits or barren cells. In this study, the co-registered cells in the GIS input layers are used to constitute the input feature vector for our neural network model. We use only 268 cells in this experiment in which 187 cells are used for training and 81 cells are used for testing. For training data, we have 85 deposit cells and 102 barren cells. For testing data, we have 35 deposit cells and 46 barren cells.

4.2 Experimental Methodology and Results

Two feed-forward backpropagation neural networks are created in this experiment. The first neural network is used as the truth NN to predict degree of favourability for deposit and another network is used as the falsity NN to predict degree of likelihood for barren. Both networks contain ten input-nodes, one output node, and one hidden layer constituting of 20 neurons. The same parameter values are applied to the two networks and both networks are initialized with the same random weights. The only difference is that the target values for the falsity NN are equal to the complement of the target values used to train the truth NN.

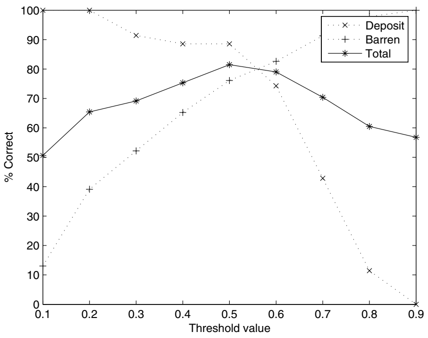
In order to estimate uncertainty of type error for the test data set, errors produced from the truth NN are plotted in the input feature space. In this study, only 60 patterns from the input training data are plotted in the input feature space because of memory limitations of the computer used in the experiment. A multidimensional interpolation [8] is then used to estimate uncertainty of type error for the test data set in the prediction of degree of favourability for deposit. We use multidimensional nearest neighbour interpolation function in Matlab to interpolate these errors. The estimation of uncertainty of type error for the prediction of degree of likelihood for barren is also calculated using the same technique as the error estimation for the deposit prediction.

In order to calculate uncertainty of type vagueness, which is the indeterminacy-membership value, equation 2 is used to compute this kind of uncertainty for each pattern in the test data set. After we created the three memberships: truth-membership, indeterminacy-membership, and false-membership for each pattern, these three memberships are then constituted into an interval neutrosophic set.

After the three memberships are determined for each pattern in the test data sets, the next step is to classify each pattern into deposit or barren cells. The truth-memberships, the false-memberships, and their uncertainties of type error are combined into a single output used for the binary classification. The dynamic combination can be computed using equation 3. After that, the classification is done by comparing each dynamic combination output to a threshold value. In this paper, threshold values are ranged from 0.1 to 0.9 in steps of 0.1. These threshold values are tested with each output to seek for the best threshold value

Table 1. Classification accuracy for the test data set obtained by applying a range of threshold values to the output of dynamic weighted combination among truth-membership, false-membership, and uncertainties of type error. (Right: graphical representation of data in this table)

Threshold value	Deposit %correct	Barren %correct	Total %correct
0.1	100.00	13.04	50.62
0.2	100.00	39.13	65.43
0.3	91.43	52.17	69.14
0.4	88.57	65.22	75.31
0.5	88.57	76.09	81.48
0.6	74.29	82.61	79.01
0.7	42.86	91.30	70.37
0.8	11.43	97.83	60.49
0.9	0.00	100.00	56.79



that produces the best accuracy in the classification. For each cell c_i in the classification, if the truth-membership value is greater than the threshold value then the cell is classified as a deposit. Otherwise, it is classified as barren. Table 1 shows classification accuracy for the test data set obtained by applying a range of threshold values to the output of dynamic combination. We found that the maximum of the total correct cell in the classification is 81.48 percent. Hence, the optimal threshold value used in this classification is determined to be 0.5.

In this paper, we do not consider the optimization of the prediction, but our purpose is to test a new approach that provides an estimate of uncertainty in the prediction. We compare our classification results with those obtained using the traditional method for binary classification. In the traditional approach, only truth-membership values are used in the comparison. If the cell has the truth-membership value greater than the threshold value then the cell is classified as deposit. Otherwise, the cell is classified as barren. Table 2 shows classification accuracy for the test data set obtained by applying a range of threshold values to the only truth-membership values. The maximum of the total correct cell in the traditional classifications is 80.25 percent. Therefore, the optimal threshold value used in this traditional classification is determined to be 0.5.

The results from our proposed classification using the dynamic combination represent 1.23 percent improvement over those obtained using the traditional classification applied only the truth-membership values. Table 3 shows samples of individual predicted cell types and their uncertainties of type error and vagueness resulted from our proposed model for the test data set. The individual predicted cell types for the traditional approach are also shown in this table in the last column. These samples are shown that our proposed model has an advantage of quantification of uncertainty in the prediction. For example, the actual cell type for the cell in the first row of this table is a deposit cell, but it is predicted to be a barren cell. The traditional approach cannot explain about uncertainty in this prediction, but our approach can explain that the cell is predicted to be a

Table 2. Classification accuracy for the test data set obtained by applying a range of threshold values to the truth-membership values. (Right: graphical representation of data in this table)

Threshold value	Deposit %correct	Barren %correct	Total %correct
0.1	100.00	30.43	60.49
0.2	94.29	47.83	67.90
0.3	91.43	56.52	71.61
0.4	91.43	63.04	75.31
0.5	88.57	73.91	80.25
0.6	77.14	80.43	79.01
0.7	54.29	89.13	74.07
0.8	22.86	95.65	64.19
0.9	2.86	100.00	58.02

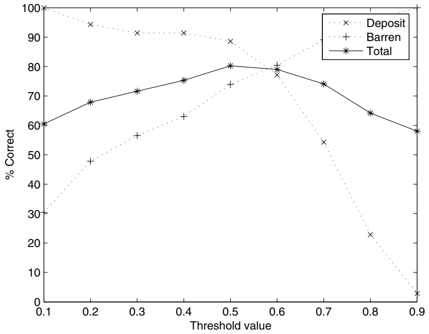


Table 3. Sample outputs from the proposed model for the test data set (columns 2-7) together with classifications based on dynamic combination (column 8) and traditional classifications based on truth-membership values (column 9)

Actual Cell Type	$T(c_i)$	$e_t(c_i)$	$F(c_i)$	$e_f(c_i)$	$I(c_i)$	Dynamic Combination $O(c_i)$	Predicted Cell Type $O(c_i) > 0.5$	Predicted Cell Type $T(c_i) > 0.5$
Deposit	0.40	0.04	0.70	0.16	0.70	0.35	Barren	Barren
Deposit	0.87	0.15	0.26	0.24	0.39	0.81	Deposit	Deposit
Deposit	0.69	0.14	0.65	0.14	0.95	0.53	Deposit	Deposit
Deposit	0.84	0.51	0.23	0.70	0.39	0.81	Deposit	Deposit
Barren	0.07	0.09	0.70	0.04	0.37	0.19	Barren	Barren
Barren	0.57	0.28	0.54	0.23	0.97	0.51	Deposit	Deposit
Barren	0.43	0.28	0.46	0.23	0.97	0.49	Barren	Barren
Barren	0.51	0.13	0.57	0.53	0.94	0.48	Barren	Deposit

barren cell with the uncertainty of 70 percent. Hence, the decision-maker can use this information to support the confidence in decision making.

Considering the last row of this table, the actual cell type for this cell is barren. Using the traditional approach, this cell is classified as a deposit which is a wrong prediction and there is no explanation of uncertainty in the prediction for this cell. Using our approach, this cell is classified as a barren, which is correct. We also know that the cell is barren with the uncertainty of 94 percent. We can see that uncertainty of type error in the prediction can enhance the classification. Therefore, the combination among the truth-membership, false-membership, and their uncertainties of type error gives the more accuracy in prediction.

5 Conclusions and Future Works

This paper represents a novel approach for mineral deposit prediction. The prediction involves ten GIS input data layers, two neural networks, an interval neutrosophic set, and a multidimensional interpolation. The co-register cells from GIS data are applied into two neural networks to produce the degrees of favourability for deposits (truth-membership) and the degrees of likelihood for barrens (false-membership). Two types of uncertainty in the prediction are estimated. These two kinds of uncertainty are error and vagueness. Estimated errors are computed using a multidimensional interpolation. Vagueness is calculated as the different between the truth- and false-membership values for each cell. This paper represents vagueness as the indeterminacy-membership. These three memberships are formed into an interval neutrosophic set. The goal of this paper is to quantify uncertainty in mineral prospectivity prediction. The more we know uncertainty information, the more certainty in decision making. In the future, we will apply this model to bagging and boosting neural networks.

References

1. Brown, W.M., Gedeon, T.D., Groves, D.I., Barnes, R.G.: Artificial neural networks: A new method for mineral prospectivity mapping. Australian Journal of Earth Sciences **47** (2000) 757–770
2. Skabar, A.: Mineral potential mapping using feed-forward neural networks. In: Proceedings of the International Joint Conference on Neural Networks. Volume 3. (2003) 1814–1819
3. Fung, C., Iyer, V., Brown, W., Wong, K.: Comparing the performance of different neural networks architectures for the prediction of mineral prospectivity. In: the Fourth International Conference on Machine Learning and Cybernetics (ICMLC 2005), Guangzhou, China (2005) 394–398
4. Duckham, M., Sharp, J.: Uncertainty and geographic information: Computational and critical convergence. In: Representing GIS. John Wiley, New York (2005) 113–124
5. Fisher, P.F.: Models of uncertainty in spatial data. In: Geographical Information Systems: Principles, Techniques, Management and Applications. 2 edn. Volume 1. John Wiley, Chichester (2005) 69–83
6. Wang, H., Madiraju, D., Zhang, Y.Q., Sunderraman, R.: Interval neutrosophic sets. International Journal of Applied Mathematics and Statistics **3** (2005) 1–18
7. Wang, H., Smarandache, F., Zhang, Y.Q., Sunderraman, R.: Interval Neutrosophic Sets and Logic: Theory and Applications in Computing. Neutrosophic Book Series, No.5. http://arxiv.org/abs/cs/0505014 (2005)
8. MATHWORKS: MATLAB Mathematics Version 7. The MathWorks Inc., Natick (2004)

Thermal Deformation Prediction in Machine Tools by Using Neural Network

Chuan-Wei Chang[1], Yuan Kang[1,2], Yi-Wei Chen[1,3],
Ming-Hui Chu[3], and Yea-Ping Wang[3]

[1] Department of Mechanical Engineering,
Chung Yuan Christian University, Chung Li 320, Taiwan, R.O.C.
yk@cycu.edu.tw
[2] R&D Center for Membrane Technology,
Chung Yuan Christian University, Chung Li 320, Taiwan, R.O.C.
[3] Department of Automation Engineering,
Tung Nan Institute of Technology, Taipei 222, Taiwan, R.O.C.

Abstract. Thermal deformation is a nonlinear dynamic phenomenon and is one of the significant factors for the accuracy of machine tools. In this study, a dynamic feed-forward neural network model is built to predict the thermal deformation of machine tool. The temperatures and thermal deformations data at present and past sampling time interval are used train the proposed neural model. Thus, it can model dynamic and the nonlinear relationship between input and output data pairs. According to the comparison results, the proposed neural model can obtain better predictive accuracy than that of some other neural model.

Keywords: Feed-forward neural network, Thermal deformation, Neural Prediction model.

1 Introduction

Thermal deformation is one of the major error sources of cutting working piece, that due to the temperature variation and non-uniform distribution characteristic. It will cause 40-70% error during cutting process in the machine tools [1]. The improvement of the finishing accuracy with error compensation by software method is very useful and efficient. The thermal deformation error can be compensated by this way without changing the design of original structure and mechanism.

The software method utilizes a mathematical model based on the measurement of temperatures to predict thermal deformations. There are some basic papers have been proposed. Donmez et al. [2] and Chen et al.[3] have applied multiple regression analysis (MRA) to predict thermal deformations for turning lathe and horizontal machine center. The MRA prediction model can be built without difficult algorithms, but it is a linear and static model. This model cannot obtain the nonlinear and dynamic relationship between the measurement temperatures and thermal deformations. The feed-forward neural network (FNN) can map the nonlinear

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 850 – 859, 2006.

relationship by the training with back-propagation algorithm [4]. It is the reasonable to use a neural network to build the thermal deformation prediction model. Hattori et al. [5] used neural network with back-propagation algorithm to modify the error prediction method for a vertical milling machine. Chen [6] and Baker et al. [7] compared the prediction errors of neural network and MRA model for a three-axis horizontal machining center and CNC machine tool. In both studies, their results show that neural network can obtain more accurate prediction than MRA method. However, these models only consider static relationship between inputs and outputs data. Thus, Yang and Ni [8] used a dynamic model to predict thermal deformation for a horizontal machining center, and obtain the better prediction result than the conventional neural prediction models.

In order to obtain the nonlinear and dynamic mapping with neural network, this study proposed a dynamic neural network model, which consists of a multiple time interval at the present and past time. This model can obtain the static, dynamic and nonlinear relationship between inputs and output data with different sampling interval. Thus, the prediction accuracy can be improved. The accuracy of an actual grinding machine with the proposed method is compared with that with conventional neural network, and the feasibility and the improvement of prediction accuracy is investigated.

2 Dynamic Neural Network Modeling

The conventional three layers of feed-forward neural network as shown in Fig. 1, this model is trained with the thermal deformations and measured temperatures at the same time interval. Therefore, it works as a static model. However, thermal deformations do not only influence by the temperatures at the present time but also by the temperatures and thermal deformations at the past time. Thus, the conventional neural network model is not sufficient for nonlinear dynamic mapping. The dynamic neural network shown in Fig. 2 can describe the dynamic relationship between the inputs and outputs data. At the nth sampling time interval, the inputs of dynamic neural network are the measured temperatures x_j from the (n-q)th to nth sampling time and the neural network outputs $\hat{y}$ from the (n-p)th to (n-1)th sampling time.

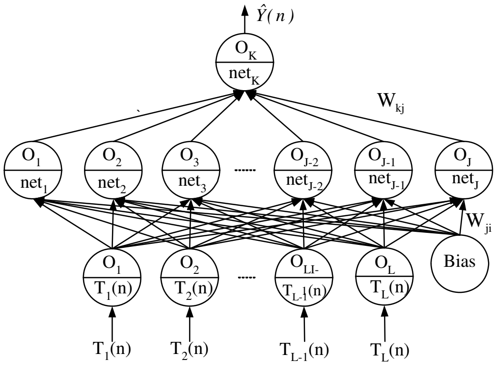


Fig. 1. Conventional neural network

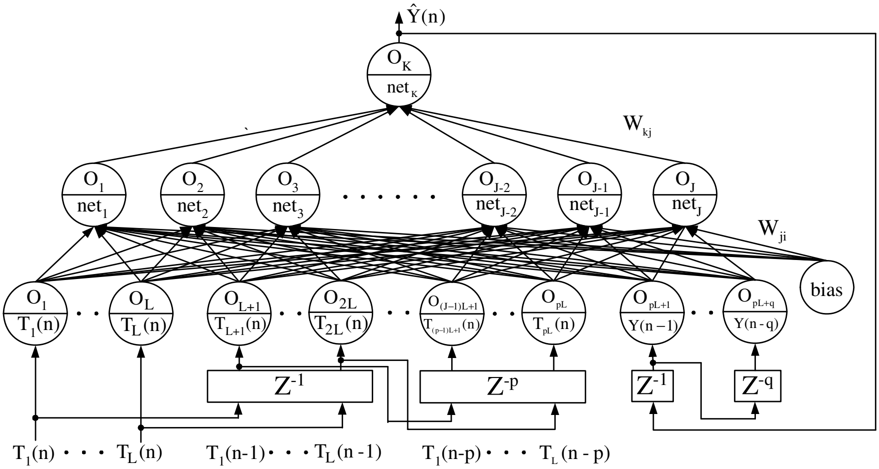


Fig. 2. Dynamic neural network

The training structure of neural network prediction model is shown in Fig. 3. In the training phase, a constant K_1 is appropriately determined to normalize the input data between 0.1 and 0.9. The connective weights between the output and hidden layer and between hidden and input layer are update by back-propagation algorithm. At the nth sampling time interval, the neural network inputs of temperatures are normalized by

$$\overline{X}_i(n) = a + (\frac{X_i(n) - X_{i,min}}{X_{i,max} - X_{i,min}})(b-a) \quad i = 1,2,\cdots,L-1,L \tag{1}$$

where $X_i(n)$ is the *ith* measured value of temperatures at the nth sampling time interval and the maximum and minimum values expressed by suffix symbol of *max* and *min* , respectively.

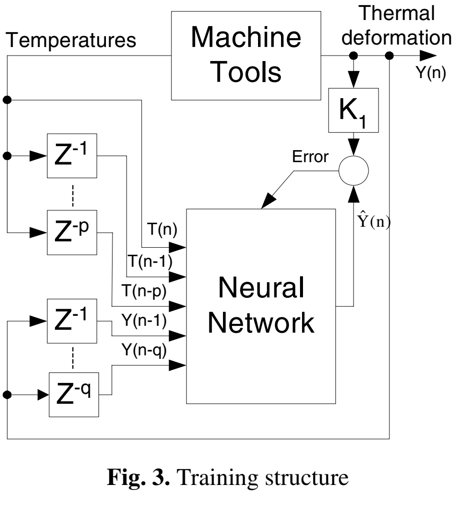


Fig. 3. Training structure

After normalization by equation (1), the inputs of ith node in the input layer can be defined by

$$Net_i(n) = T_i(n), T_i(n-1), \cdots, T_i(n-p), \hat{Y}(n-1), \hat{Y}(n-2), \cdots, \hat{Y}(n-q) \tag{2}$$

where $T_i(n)$, $T_i(n-1)$ and $T_i(n-p)$ are the ith measured temperatures at the nth, (n-1)th and (n-p)th sampling time intervals. $\hat{Y}(n-1)$, $\hat{Y}(n-2)$ and $\hat{Y}(n-q)$ are the measured thermal deformation at the (n-1)th, (n-2)th and (n-q)th sampling time interval. The initial values are

$$T(n-p) = \begin{cases} 0, & n \leq p \\ temperature, & otherwise \end{cases}$$

and

$$\hat{Y}(n-q) = \begin{cases} 0, & n \leq q \\ predicted\ \ value, & otherwise \end{cases}$$

The output of the ith node in the input layer is

$$O_i(n) = f\left(Net_i(n)\right) = tanh(\gamma Net_i(n)) \tag{3}$$

The net input of the jth node in the hidden layer is

$$Net_j(n) = \sum W_{ji}(n) O_i(n) + \theta \quad j = 1, 2, \cdots, J-1, J \tag{4}$$

where $\mathrm{W_{ji}(n)}$ are the weights between the input and hidden layers, and θ is the bias.

The jth neuron output of the hidden layer is

$$O_j(n) = f\left(Net_j(n)\right) = tanh(\gamma Net_j(n)) \tag{5}$$

where γ>0. The net input and output of the kth node in the output layer are

$$Net_k(n) = \sum_{j=1}^{J} W_{kj}(n) O_j(n) \tag{6}$$

and

$$\hat{Y}_k(n) = f\left(Net_k(n)\right) = tanh(\gamma Net_k(n)) \tag{7}$$

At the nth sampling time interval, the error energy function is defined as

$$E(n) = \frac{1}{2}\left(Y(n) - \hat{Y}(n)\right)^2 \tag{8}$$

where $Y(n)$ is the actual deformation, which is obtained by the measurement of thermal deformation and then multiplied by a constant K_1, $\hat{Y}(n)$ is the prediction value of neural network. In training phase, the neural network weights are updated during the sampling time interval from the nth to the (n+1)th according to:

$$\Delta W(n) = W(n+1) - W(n) = -\eta \frac{\partial E(n)}{\partial W(n)} \tag{9}$$

where η is denoted as learning rate.

The weights update quantities between the output and hidden layer and between hidden and input layer can be determined by following iterative:

$$\frac{\partial E(n)}{\partial W_{ji}(n)} = \frac{\partial E(n)}{\partial Net_j(n)} \frac{\partial Net_j(n)}{\partial W_{ji}(n)} = \delta_j(n) O_i(n) \tag{10}$$

$$\frac{\partial E(n)}{\partial W_{kj}(n)} = \frac{\partial E(n)}{\partial Net_k(n)} \frac{\partial Net_k(n)}{\partial W_{kj}(n)} = \delta_k(n) O_j(n) \tag{11}$$

where $\delta_j(n) = \delta_k(n) W_{kj}(n) f'(Net_j(n))$ and $\delta_k(n) = -\left(Y(n) - \hat{Y}(n)\right) f'(Net_k(n))$.

As the error energy E is less than a specified value, then the training iteration is finished. The prediction model will be built consequently.

3 Experimental Setup

A machine tool has been used to experiment with 1645×2140×2181 mm of size specification and 152×355×305mm of working space. The worktable motion in the X-axis is driven by hydraulic mechanism and controlled by a directional control valve. Therefore, this working axis cannot be precisely controlled by servo command. The worktable motions in Y axis and Z axis are driven by AC servomotors. Thus, the prediction of the thermal deformations will be used in the Y- and Z-axis.

The measurement system in experiments for this study is shown in Fig. 4. The thermal sensors and non-contact displacement sensors are used to measure the temperature and thermal deformations, respectively. The measurement data is process by the professional small signal amplifier and the data acquisition (DAQ) of NI PCI 6024E. These data are analyzed by the soft pachage Labview with an AMD Duron 800, PC system.

There are thirteen locations for the temperature measurement as shown in Fig. 5 is chosen by trial and error empirically. Which locate at spindle shield, the column, and slideways of Y- and Z-axis, hydraulic oil tank, respectively, and a sensor is located nearby from this machine for the measurement of environment temperatures.

The displacement sensors are capacitance type with accuracy of 0.1 μm and mounted on two precise vises as shown in Fig. 6, which are used to measure the relative displacements the spindle end in directions of Y- and Z-axis.

During the experiment phase, the spindle runs at constants speed of 3600 rpm for eight hours. The spindle end is fixed at –75mm from the mechanical origin. The worktable mores moves in X- and Z-axis with constant speed of 1000 mm/min repeatedly from the mechanical origin to –360mm and –120mm, respectively. One

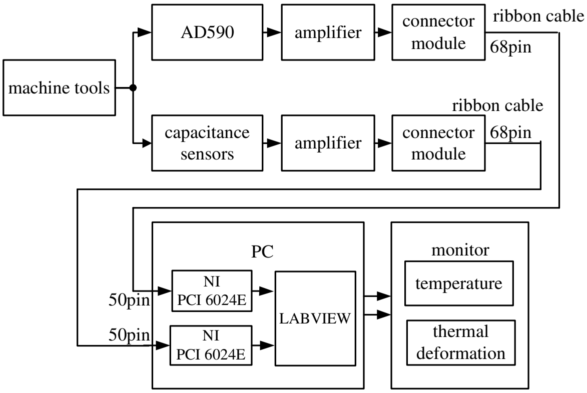


Fig. 4. Measurement system

minute is selected to be sampling time for temperature and deformation data. The experiment runs for two times. The measurement of the ambient temperature shows that the variation is from 3 °*C* to 10 °*C* during experiment times, the spindle temperature increases from 5 °*C* to 10 °*C* , the temperature of hydraulic tank increases from 15 °*C* to 25 °*C* , the slideway temperature of Z axis increases from 3 °*C* to 5 °*C* , and the temperatures of ball screw in Y and Z axes increases from 4 °*C* to 6 °*C* . The thermal deformations of the spindle end are measured from –8 μm to 0 μm in the Y axis and from –13 μm and 5 μm in the Z axis. The measurement results of thermal deformation for the first experimental in Y- and Z- axes are shown in Fig. 7.

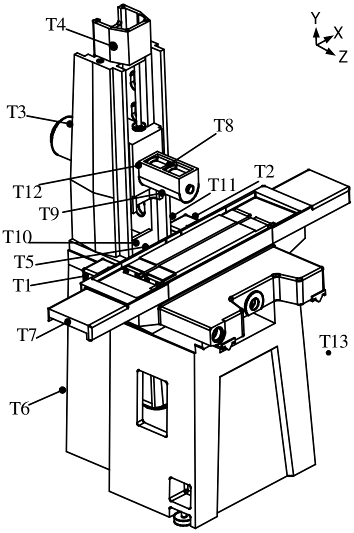


Fig. 5. Locations of temperature measurements

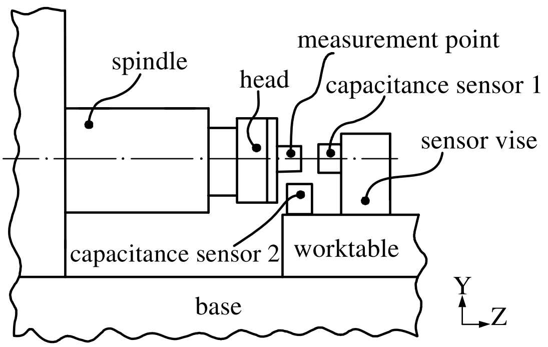


Fig. 6. Displacement measurements of thermal deformation

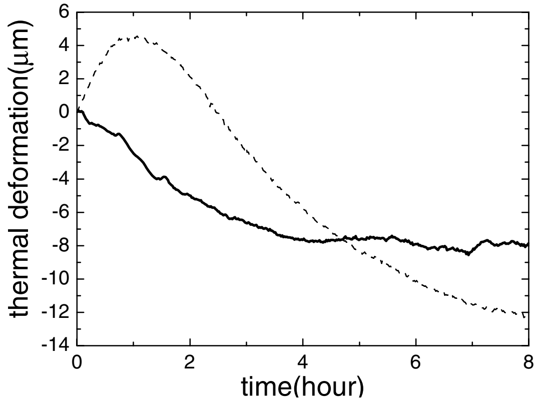


Fig. 7. Thermal deformation of measurement results (Y axis: ————; Z axis: — — — —)

4 Case Study

The measurement data of the first experiment are used to train the prediction model. The hidden layer of neural network has 30 nodes. The normalization range is between 0.1 and 0.9, this range is the same as the ranges of coefficients a and b of equation (1). The initial weights of neural network are randomly generated between –0.5 and +0.5. The modified model for prediction of thermal deformations is built by training 5,000 epoch.

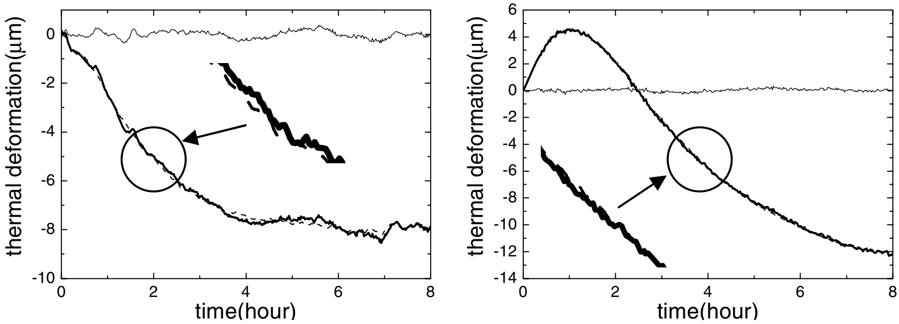
In this study, the temperature measurement data and previous neural network outputs are used as inputs of neural network, which are p=1, q=3, respectively. The neural network utilizes one output neuron, its outputs utilize as the prediction values of thermal deformations for Y axis or Z axis. In the prediction phase, the thermal deformation prediction values at the past sampling time interval are the neural network inputs, it is different of training phase.

Both the maximum and root mean square (RMS) prediction errors are used to denote the prediction accuracies. Both of the training and prediction phases use the same data of first experiment, the RMS errors in Y and Z axes are 0.15 μm and 0.11 μm, respectively. The maximum errors in Y and Z axes are 0.38 μm and 0.34 μm, respectively. The training results for both axes are shown in Fig. 8. The maximum training error can be less than 0.4 μm in both axes.

The prediction results of time history in second experiment are shown in Fig. 9. The RMS and maximum errors of prediction results for these methods are listed in Table 1. All values of errors of the dynamic model are smaller than static model. The RMS errors in Y and Z axes are 1.37 μm and 0.57 μm, respectively. The maximum errors in Y and Z axes are 2.98 μm and 1.09 μm, respectively. The dynamic neural network can improve the accuracy, and be more accurate than conventional neural network model.

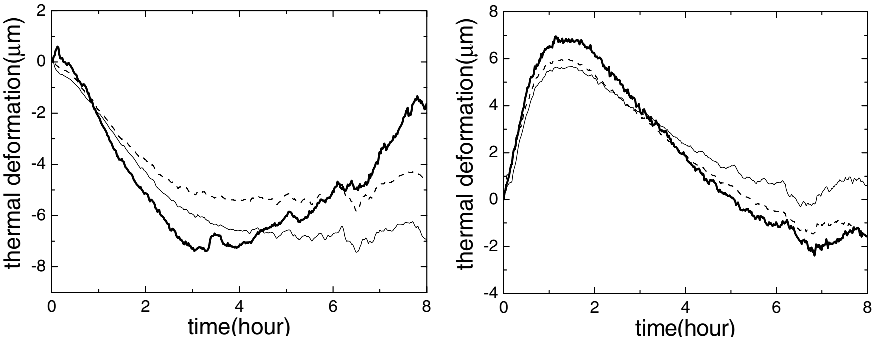
Table 1. Prediction error (μm)

		Conventional NN	Dynamic NN
Y axis	RMS error	1.83	1.37
	Maximum error	5.30	2.98
Z axis	RMS error	1.33	0.57
	Maximum error	2.54	1.08



(a) Y-axis (b) Z-axis

Fig. 8. Prediction model of training results for Y-and Z-axis (experimental measurement: ———; dynamic neural network: — — — —; and prediction error: ———)



a) Y-axis (b) Z-axis

Fig. 9. Thermal deformation of prediction results for Y-and Z-axis (experimental measurement: ————; dynamic neural network: — — — —; and conventional neural network: ————)

5 Conclusion

This study develops a dynamic neural network model to predict thermal deformation in machine tools. The dynamic and nonlinear relationship based on the input and output data pairs of different sampling time interval can be emulated by the performance of neural network. The experimental results show that the dynamic neural network model is more accurate than the conventional neural network. The maximum prediction errors can be less than 3 μm and 2 μm for Y-and Z-axis, respectively. Thus, the dynamic neural network model is proven to be useful for industrial applications.

Acknowledgments. This study was supported by grant number NSC94-2745-E-033-005-URD from National Science Council and thanks for the center-of-excellence program supported by the Ministry of Education of R.O.C..

References

1. Bryan, J., "International Status of Thermal Error Research," Annals of the CIRP, Vol. 39 (1990) 645-656.
2. Donmez, M. A., Blomquist, R. J., Hocken, R. J. Liu, C. R. and Barash, M. M. "A General Methodology for Machine Tool Accuracy Enhancement by Error Compensation, Precision Engineering," Vol. 8, No. 4, (1986) 187-195.
3. Chen, J. S., Yuan, J. X., Ni, J. and Wu, S. M., "Real-time Compensation for Time-variant Volumetric Errors on a Machining Center," Journal of Engineering Industry, Vol. 115, (1993) 472-479.
4. Freeman, J. A. and Skapura D. M. "Neural Networks Algorithms, Applications, and Programming Techniques," Addison Wesley, New York, (1992).
5. Hattori, M., Noguchi, H., Ito, S., Suto, T. and Inoue, H., "Estimation of Thermal Deformation in Machine Tools Using Neural Network Technique," Journal of Materials Processing Technology, Vol. 56, (1996) 765-772.

6. Chen, J. S., “Neural Network-based Modelling and Error Compensation of Thermally-induced Spindle Errors,” The International Journal of Advanced Manufacturing Technology, Vol.12, (1996) 303-308.
7. Baker, K., Rao, B. K. N. Hope, A. D. and Noroozi, S., “Performance Monitoring of a Machining Centre,” IEEE, Instrumentation and Measurement Technology Conference, (1996) 853-858.
8. Yang H. and Ni, J., “Dynamic Modeling for Machine Tool Thermal Error Compensation,” Journal of Manufacturing Science and Engineering, Vol. 125, (2003) 245-254.

Fuzzy Time Series Prediction Method Based on Fuzzy Recurrent Neural Network

Rafik Aliev[1], Bijan Fazlollahi[2], Rashad Aliev[3], and Babek Guirimov[1]

[1] Azerbaijan State Oil Academy
raliev@iatp.az
[2] Georgia State University
dscbbf@langate.gsu.edu
[3] Eastern Mediterranean University
rashad.aliyev@edu.emu.tr

Abstract. One of the frequently used forecasting methods is the time series analysis. Time series analysis is based on the idea that past data can be used to predict the future data. Past data may contain imprecise and incomplete information coming from rapidly changing environment. Also the decisions made by the experts are subjective and rest on their individual competence. Therefore, it is more appropriate for the data to be presented by fuzzy numbers instead of crisp numbers. A weakness of traditional crisp time series forecasting methods is that they process only measurement based numerical information and cannot deal with the perception-based historical data represented by fuzzy numbers. Application of a fuzzy time series whose values are linguistic values, can overcome the mentioned weakness of traditional forecasting methods. In this paper we propose a fuzzy recurrent neural network (FRNN) based fuzzy time series forecasting method using genetic algorithm. The effectiveness of the proposed fuzzy time series forecasting method is tested on benchmark examples.

1 Introduction

Forecasting activities on the basis of prediction from time series play an important role in different areas of human activity, including weather forecasting, economic and business planning, inventory and production control, etc. Many available data in such real-world systems, where a human plays the basic decision maker role, are linguistic values or words with fuzzy meaning. The main advantage of using fuzzy approach is to apply human expertise throughout the forecasting procedure. This type of time series significantly differs from traditional time series and the methods of the latter are not applicable in this case. So we deal with a new class of time series, a fuzzy time series, whose values are linguistic values. There are several approaches to modeling fuzzy time series.

In [20-23], fuzzy time series models are proposed and their applications to forecasting problems are considered. Some extension of fuzzy relational model based time series is analyzed in [26]. Applications of fuzzy time series are considered in [9,25]. Modification of fuzzy time series models for forecasting of university enrollments is discussed in [10]. In [4] a fuzzy fractal method for forecasting financial and economic time series is described.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 860 – 869, 2006.

All the works on fuzzy time series mentioned above are based on fuzzy relational equations [29]. The forecasting methods based on fuzzy relational equations suffer from a number of shortcomings among which the main ones are computational complexity, difficulty of choosing an optimal or near-optimal fuzzy implication, difficulty of training and adaptation of the rule base (relational matrices) and its parameters and others. These drawbacks lead to difficulties in reaching the desirable degree of forecasting accuracy. Application of fuzzy neural network for time series forecasting can overcome these weaknesses [1]. Note also that recurrent neural networks are found to be very effective and efficient technique to use in many dynamic or time series related applications. Paper [12], for example, proposes a recurrent fuzzy neural network for identification and control of dynamic systems.

In this paper we propose a FRNN based fuzzy time series forecasting method which allows application of human expertise throughout the forecasting procedure effectively. This method is characterized by the less computational complexity, learning by experiments, and adaptability.

The rest of this paper is organized as follows. Section 2 gives the statement of the time series forecasting problem using Fuzzy Recurrent Neural Networks (FRNN). Section 3 describes the suggested forecasting system's structure and operation principles. Section 4 contains description of computational experiments with the FRNN based time series forecasting system. Various benchmark problems are used for system performance validation purposes and comparisons with other systems are provided. Section 5 is the conclusion of the paper. The references used are listed following section 5.

2 Statement of the Problem

Suppose that at various time instant t we are presented with perception based information y_t described by fuzzy sets. Formally, the fuzzy n-order time series problem can be represented as:

$$y_{t+1} = F(y_t, y_{t-1},..., y_{t-n+1}), \tag{1}$$

where F is fuzzy set valued mapping of values $y_t, y_{t-1},..., y_{t-n+1} \in \varepsilon^n$ from ε^n into $y_{t+1} \in \varepsilon^1$ to be estimated, ε^n and ε^1 are spaces of fuzzy sets, y_t is fuzzy valued data at time interval t, y_{t+1} is fuzzy valued forecasted value for time interval t+1.

As it was stated in [1,13] fuzzy neural networks are universal approximators, and hence can be used to construct fuzzy set valued mapping F in (1). In a simple case, we need a FNN with one hidden layer, n input and one output nodes which can express the relationships:

$$\hat{y}_{t+1} = \hat{F}_{NN}(y_t, y_{t-1},..., y_{t-n+1}), \tag{2}$$

where an estimate $\hat{F}_{NN}$ for F is constructed from a large class of fuzzy neural network based mappings. In its turn $\hat{F}_{NN}$ is determined by fuzzy weights of neuron connections, fuzzy biases, and neuron activation functions. The problem is in adjusting the weights to minimize a cost function $E(F, \hat{F}_{NN})$ (for instance, as fuzzy hamming distance), defined on the basis of (1) and (2) and representing distance measure between the fuzzy neural network output and the desired output pattern.

The system may use also some additional time-dependent factors if it influences the value of the forecasted variable:

$$\hat{y}_{t+1} = \hat{F}_{NN}(y_t, y_{t-1}, ..., y_{t-n+1}, u_t, ..., u_{t-1}, u_{t-m+1}), \tag{3}$$

where u_t is the value of an additional (second) factor at time interval *t*.

Using benchmark tests, it will be shown that the forecasting error of this method is significantly smaller than that of existing fuzzy time series approaches [6,10,20-22,26].

3 The FRNN Based Forecasting System and Its Operation

The structure of the forecasting system on the basis of fuzzy recurrent neural network for the realization of (2)-(3) is presented in figure 1. For multi-lag forecasting the FRNN output signal is fed-back to the network input each time producing the forecast for a next time. For example, the output signal $\hat{y}_{t+1}$, produced for an actual time series $y_{t-n+1}, ..., y_{t-1}, y_t$, applied back to the input would produce an output $\hat{y}_{t+2}$ giving an approximation for y_{t+2}.

The neurons in the layers 1 to layer *L* are dynamic and compute their output signals as follows:

$$z_i^l(t) = F\left(\theta_i^l + \sum_j x_j^l(t) w_{ij}^l + \sum_j z_j^l(t-1) v_{ij}^l\right), \tag{4}$$

where $x_j^l(t)$ is *j*-th fuzzy input to the neuron i at layer l at the time step *t*, $z_i^l(t)$ is the computed output signal of the neuron at the time step *t*, w_{ij} is the fuzzy weight of the connection to neuron *i* from neuron *j* located at the previous layer, θ_i is the fuzzy bias of neuron *i*, and $z_j^l(t-1)$ is the activation of neuron *j* at the time step (*t*-1), v_{ij} is the recurrent connection weight to neuron *i* from neuron *j* at the same layer.

In general, the network may have virtually any number of layers. We number the layers successively from 0 (the first or input layer) to *L* (last or output layer). The neurons in the first (layer 0) layer are only distributing the input signal without modifying the values:

$$z_i^0(t) = x_i^0(t), \tag{5}$$

The network may have one input or more (if exogenous inputs are used or several historical data series are fed in parallel as in case of non-recurrent networks). For instance, in a time series forecasting system with only one forecasted variable input and one exogenous input (both fed by consecutive historical data), the FRNN input

$x_0^0(t)$ will represent the time series element y_t, input $x_1^0(t)$ will represent the additional factor input u_t, and the FRNN output $z^L(t)$ will represent the forecasted time series element y_{t+1}.

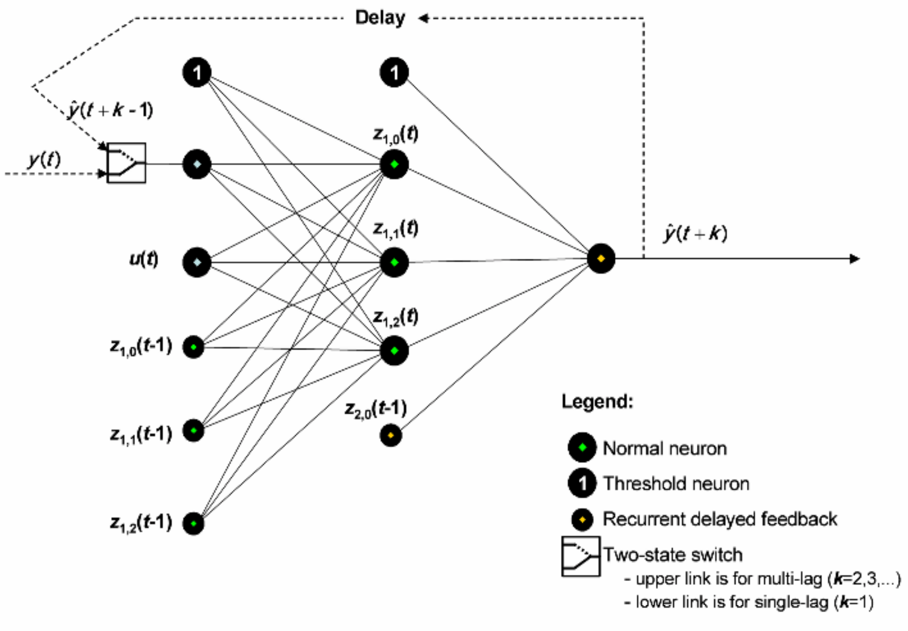


Fig. 1. The structure of a simple FRNN

The activation *F* for a total input to the neuron s is calculated as

$$F(s) = \frac{s}{1 + |s|}, \tag{6}$$

So, the output of neuron *i* at layer *l* is calculated as follows:

$$z_i^l(t) = \frac{\theta_i^l + \sum_j x_j^l(t) w_{ij}^l + \sum_j z_j^l(t-1) v_{ij}^l}{1 + \left| \theta_i^l + \sum_j x_j^l(t) w_{ij}^l + \sum_j z_j^l(t-1) v_{ij}^l \right|}, \tag{7}$$

All fuzzy signals and connection weights and biases are general fuzzy numbers that with any required precision can be represented as

$$T(L_0, L_1, ..., L_{n-1}, R_{n-1}, R_{n-2}, ..., R_0)$$

In case of *n*=2 (used in experiments) a fuzzy number T(L0, L1, L2, R2, R1, R0) is a fuzzy set of three intervals (α-cuts): [L0,R0] (α=0), [L1,R1] (α=0.5), and [L2,R2]

(α=1). n is set to a larger value to increase the accuracy of fuzzy operations. The details of arithmetic operations with the used fuzzy number representation can be found in [1] and [3].

The problem now is adjusting the weight matrices to minimize a cost function (the FRNN error performance index) $E_{tot}(y_p, y_p^{des})$ defined by (8) and representing a distance measure between the neural network output and the desired output pattern:

$$E_{tot} = \sum_{p} D(y_p, y_p^{des}) , \tag{8}$$

where E_{tot} is the total error performance index for all learning data entries p.

We shall assume Y is a finite universe $Y = \{y_1, y_2, ..., y_n\}$; D is an error function such as the distance measure between two fuzzy sets, the desired y_p^{des} and the computed y_p outputs. The efficient strategy is to consider the difference of all the points of the used general fuzzy number. The considered distance metrics is based on Hamming distance

$$D(T1,T2) = \sum_{i=0}^{i=n-1} k_i \mid L_{T1i} - L_{T2i} \mid + \sum_{i=0}^{i=n-1} k_i \mid R_{T1i} - R_{T2i} \mid , \tag{9}$$

where $D(T1,T2)$ is the distance measure between two fuzzy numbers $T1(\,y_p^{des}\,)$ and $T2(\,y_p\,)$; $0 \le k_0 \le k_1 ... \le k_{n-2} \le k_{n-1}$ are some scaling coefficients. For example, for n=2, an effective distance metrics could be:

$$D(T1,T2) = k_0|L_{T10}-L_{T20}| + k_1|L_{T11}-L_{T21}| + k_2|L_{T12}-L_{T22}| +$$
$$+ k_0|R_{T10}-R_{T20}| + k_1|R_{T11}-R_{T21}| + k_2|R_{T12}-R_{T22}|,$$

where k_0=1/7; k_1=2/7; k_2=4/7.

In this paper we use genetic algorithm-based learning algorithm for FRNN with fuzzy inputs, fuzzy weights and biases, and fuzzy outputs suggested in [2,3]. The fitness function of a genome (or chromosome) is calculated on the basis of the total error performance index (for a particular combination of FRNN weights and thresholds) as follows:

$$f = \frac{1}{1 + E_{tot}} \tag{10}$$

The learning may be stopped once we see the process does not show any significant change in fitness value during many succeeding regenerations. In this case we can specify new mutation (and maybe crossover) probability and continue the process. If the obtained total error performance index or the behavior of the obtained network is not desired, we can restructure the network by adding new hidden neurons, or do better sampling (fuzzification) of the learning patterns.

The genomes in the current population undergo specific genetic operators, which leads to a change in population: new child genomes (offsprings) are produced. To rank these genomes, their fitness values are calculated. To do this, first they are converted into the network representation and the network error is calculated, and then formula (10) is applied to calculate the fitness.

During the selection processes low fitness genomes have low probability to survive and be saved into a new population for participation in future reproduction. The process is repeated iteratively. At every generation we have a solution that corresponds to a genome with the highest fitness function. The farther we go with generations the higher is the chance to find a better solution.

The detail of the used GA based fuzzy network training procedure is considered in [2,3].

4 Computational Experiments

To test the suggested FRNN based fuzzy time series forecasting method several benchmark problems are considered in this paper. Frequently time series data base and perception based information is characterized by uncertainty and imprecision. It is possible to model uncertain information, linguistic terms, and not ill-defined or imprecise data by means of fuzzy sets. In general, a fuzzy pre-processing is required to transform the measurement values extracted from the given signals into a linguistic distribution (e.g. fuzzification) [2,3]. It should be noted that fuzzy sets are used in this work for the treatment and management of imprecision and uncertainty at the various levels of the prediction system, i.e. from the measurement to the inference levels [2,3]. In all our experiments all connection weights and biases are coded as 64 bits long genes. In simulation experiments we used fuzzy numbers of type: $T(L_0, L_1, L_2, R_2, R_1, R_0)$. We used 12 bits for coding L_2 and R_2, 12 bits for coding L_1 and R_1 and 8 bits for coding L_0 and R_0. Thus, for example, if we have in total 54 adjustable parameters in the network, the genome length would be 54×64=1088 bits.

For better learning we use 100 genomes. All 100 genomes undergo the multi-point crossover and mutation operations [1-3].

Then every 90 best offspring genomes plus 10 best parent genomes make a new population of 100 genomes (we preserve best 10 parent genomes in every next generation). The selection of 100 best genomes is done on the basis of the genome fitness value.

4.1 Prediction of Electricity Consumption

We tested the suggested FRNN based approach on the electricity consumption forecasting problem considered in [8]. The FRNN constructed for this problem had 1 input, 7 hidden neurons (with recurrent links), and 1 output neuron (with recurrent link). Of all available data 80% were used for training and the rest were used for testing.

Table 1 shows the comparison of performance results given by different methods for unknown data (years 1999 to 2002). Absolute percentage errors for electricity consumption values computed for different years and the MAPE (Mean Absolute Percentage Error) on these data by three different methods are shown: by the method based on Back Propagation Feed Forward (BP) neural network, by the method based on Radial Basis (RB) neural network, and by the method based on FRNN.

Table 1. Comparison of different forecasting methods

Years	BP Network	RB Network	Our approach (FRNN)
1999	1.09	4.21	4.37
2000	0.44	5.83	0.37
2001	6.49	0.76	0.84
2002	5.69	0.28	4.75
MAPE	3.42	2.77	2.58

As can be seen the results shown by FRNN is better than the results shown by non-recurrent network based forecasting systems.

4.2 Temperature Prediction

Table 2 shows a fragment of historical data of the daily average temperature in Taipei [6]. 85% of the daily average temperature values in June, July, and September, fuzzified in advance as shown in Table 2, were used for training of FRNN.

Table 2. Average temperature in Taipei ($^{\circ}C$) in June 1996

Day	Crisp value of Temp.	Linguistic value of Temperature								
		Very-Very Low	Very Low	Low	More or less Low.	Avera-ge	More or less High	High	Very High	Very-Very High
1	26.1	0	0	0	0.933	0.067	0	0	0	0
2	27.6	0	0	0	0	0.933	0.067	0	0	0
3	29.0	0	0	0	0	0	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...
31	26.1	0	0	0	0.933	0.067	0	0	0	0

As can be seen from Table 3 the data were fuzzified by 9 linguistic terms: “Very-Very Low”, “Very Low”, “Low”, “More of Less Low”, “Average”, “More or Less High”, “High”, “Very High”, and “Very-Very High”.

The mean absolute percentage error achieved by our approach was 2.61%, RMSE=0.90. This error value is lower than the error values (ranging from 2.75% to 3.49% produced by different algorithms) obtained by the methods suggested in [6].

4.3 Forecasting Enrollments in University of Alabama

The problem of forecasting enrolments in University of Alabama is discussed in [11]. Data for this problem were taken from [11]. Various forecasting systems have been tested on this problem [5,11,20,23]. We used the suggested FRNN based approach on this problem too. The system predicts the number of enrolled students for the next year given the actual number of enrolled students in the previous year. The network had 3 layers, with one neuron in the input layer, 10 neurons in the hidden layer, and one neuron in the output layer. Of the all available historical data 70% were used for learning the system and the remaining 30% to calculate the forecasting accuracy.

The offered approach's root mean square error (RMSE=194) is smaller than the Chen's method (630.86) [5], the Song-Chissom method (421.32) [20], and the Tsai and Wu method using high order fuzzy time series (199.32) [27].

Table 3 shows forecasting error produced by different methods in mean absolute percentage error (MAPE):

Table 3. Comparison of different forecasting methods for enrollments forecasting problem

Song-Chissom [20]	Hwang-Chen-Lee [11]	Markov [23]	Our approach (FRNN)
3.15%	2.79%	2.6%	0.9%

The feasibility of the use of suggested FRNN based method to forecast time series is evident from the test.

4.4 Sunspot Prediction

The performance of FRNN was also tested on a well-known problem of sun-spot prediction [19]. The historical data for this problem were taken from the Internet. Several data sets were prepared as in [19]. The data used for training were sun-spot data from years 1700 to 1920. Two unknown prediction sets used for testing were from 1921 to 1955 (PR1) and from 1956 to 1979.

The comparison of performance of the FRNN approach with other existing methods for the datasets PR1 and PR2 is presented in Table 4 (NMSE i.e. the Normalized Mean Square Error measure is used in these experiments). The last two rows in Table 2 were obtained by networks trained on the same data sets by two different persons independently.

Table 4. Comparison of different forecasting methods for sun-spot prediction problem

Author (Method)	Number of inputs	PR1	PR2
Rementeria (AR) [18]	12	0.126	0.36
Tong (TAR) [24]	12	0.099	0.28
Subba Rao (Bilinear) [17]	9	0.079	-
DeGroot (ANN)[7]	4	0.092	-
Nowland (ANN) [14]	12	0.077	-
Rementeria (ANN) [18]	12	0.079	0.34
Waterhouse (HME) [28]	12	0.089	0.27
(FRNN-1)	1	0.066	0.22
(FRNN-2)	1	0.074	0.21

The results of the suggested FRNN approach are very good, taking into consideration the simple structure of network having only 1 input neuron.

5 Conclusions

We have proposed a fuzzy recurrent neural network based fuzzy time series forecasting method which can deal with both historical numerical and perception type data. The distinguishing features of the proposed forecasting method are: the ability to apply human expertise throughout the used forecasting information; the ability of FRNN to update the forecasting rules extracted from a dynamic data mining procedure; less computational complexity in comparison to existing fuzzy time series models due to parallel processing of perceptions and data and fast fuzzy inference based on FRNN; significantly high degree of forecasting accuracy in comparison with fuzzy time series models based on fuzzy relational equation; universal approximation and learning from time series data base.

The proposed method for forecasting fuzzy time series demonstrated a very high efficiency and performance. The developed fuzzy time series forecasting method was tested on four data sets: an electricity consumption prediction problem (obtained accuracy was 2.58%, reduced from 2.77% by an ordinary neural network), the benchmark problem of forecasting of enrolments to the University of Alabama (obtained accuracy was 0.9%, reduced from 2.6% by the best other method), a temperature prediction system (obtained accuracy was 2.61%, reduced from 2.75% by the best other method), and the well-known sun-spot forecasting problem (the obtained accuracy exceeds the accuracy by other methods).

References

1. Aliev R.A., Fazlollahi B., Aliev R.R., Soft Computing and Its Applications in Business and Economics, Springer Verlag, 2004, 450 p.
2. Aliev R.A., Fazlollahi B., Vahidov R., Genetic Algorithm-Based Learning of Fuzzy Neural Networks. Part 1: Feed-Forward Fuzzy Neural Networks, Fuzzy Sets and Systems 118, 2001, pp. 351-358.
3. Aliev R.A., Guirimov B.G., Fazlollahi B., Aliev R.R. “Genetic Algorithm-Based Learning of Fuzzy Neural Networks. Part 2: Recurrent Fuzzy Neural Networks” (submitted to Fuzzy Sets and Systems, 2006).
4. Castillo O., Melin P. A New-Fractal Approach for Forecasting Financial and Economic Time Series. J IEEE: 929-934, 2001.
5. Chen S.M. Forecasting Enrolments based on fuzzy time series, Fuzzy Sets and Systems 81, 1996, pp. 311-319.
6. Chen S.M., Hwang J.R.. Temperature Prediction Using Fuzzy Time Series. Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, Vol. 30, No. 2, 2000, pp. 263-275.
7. DeGroot, Wurtz D. Analysis of Univariate Time Series with Connectionist Nets: A Case Study of Two Classical Examples, Neurocomput., Vol. 3, 177-192, 1991.
8. Hamzacebi, C., Kutay F. Electric Consumption Forecasting of Turkey Using Artificial Neural Networks Up to Year 2000. J. Fac. Eng. Arch. Gazi Univ., Vol. 19, No 3, 227-233, 2004.
9. Hwang J.R, Chen S.M, Lee C.H. A new method for handling forecasting problems based on fuzzy time series. In: 7tln Internat. Conf. On Information Management. Chungli, Taoyuan, Taiwan, ROC, pp 312-321, 1996.
10. Hwang J.R., Chen S.M., Lee C.H. Handling forecasting problems using fuzzy time series. J Fuzzy Sets and Systems 100: 217-228, 1998.
11. Hwang J.R., Chen S.M., Lee C.H. Handling Forecasting Problems Using Fuzzy Time Series. Fuzzy Sets and Systems 100, pp. 217-228, 1998.
12. Lee C.-H., Teng C.-C. Identification and Control of Dynamic Systems Using Recurrent Fuzzy Neural Networks. IEEE Trans. on Fuzzy Systems, Vol. 8, No. 4, 349-366, 2000.
13. Liu P., Li H. Fuzzy Neural Network Theory and Applications, World Scientific, 376 p., 2004.

14. Nowland S., Hinton G. Simplifying Neural Networks by Soft Weight-Sharing, Neural Comput., Vol. 4, No. 4, 473-493, 1992.
15. Pedrycz W. Fuzzy Control and Fuzzy Systems, John Wiley, New York, 1989.
16. Pedrycz W. Neurocomputations in Relational Systems. IEEE Trans. on Pattern Analysis and Machine Intelligence, 13(3), pp. 289-297.
17. Rao S., Gabr M.M. An Introduction to Bispectral Analysis and Bilinear Time Series Models, in Lecture Notes in Statistics. Springer-Verlag, Vol. 24, 1984.
18. Rementeria S., Olabe X. Predicting Sunspots with a Self-Configuring Neural System, in Proc. 8th Int. Conf. Information Processing Management Uncertainty Knowledge-Based Systems, 2000.
19. Sfetsos, A., Siriopoulos, C. Time Series Forecasting with Hybrid Clustering Scheme and Pattern Recognition. IEEE Trans. on Systems, Man, and Cybernetics – part A: Systems and Humans, Vol. 34, No. 3, 399-405, 2004.
20. Song Q, Chissom B.S. Forecasting enrollments with fuzzy time series – part I. J Fuzzy Sets and Systems 54: 1-9, 1993.
21. Song Q, Chissom B.S. Forecasting enrollments with fuzzy time series – part II. J Fuzzy Sets and Systems 62: 1-8, 1994.
22. Song Q, Chissom B.S. Fuzzy time series and its models. J Fuzzy Sets and Systems 54: 269-277, 1993.
23. Sullivan J, Woodall W.H. A comparison of fuzzy forecasting and Markov modeling. J Fuzzy Sets and Systems 64: 279-293, 1994.
24. Tong H, Lim K.S. Threshold Autoregression, Limit Cycle and Cyclical Data. Int. Rev. Statist. Soc. B, Vol. 42, 1980.
25. Tsai C.C., Wu S.J. A Study for Second-order Modeling of Fuzzy Time Series. In: IEEE International Fuzzy Systems Conference. Seoul, Korea, 1999
26. Tsai C.C., Wu S.J. Forecasting enrollments with high-order fuzzy time series. In: IEEE 19n International Conference of the North American Fuzzy Information Processing Society, pp 196-200, 2000.
27. Tsai C.C., Wu S.J. Forecasting Enrolments with High-Order Fuzzy Time Series. in Proceedings of 19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS), pp. 196-200.
28. Waterhouse S.R., Robinson A.J. Non-Linear Prediction of Acoustic Vectors Using Hierarchical Mixtures of Experts, in Advances of Neural Information Processing Systems. Cambridge, MA: MIT Press, Vol. 7, 1995.
29. Zadeh L The concept of a linguistic variable and its application to approximate reasoning. J Information Sciences 8: 43-80, 1975.
30. Zuoyoung L., Zhenpei C., Jitao L. A model of weather forecast by fuzzy grade statistics. J Fuzzy Sets and Systems 26: 275-281, 1988.

Research on a Novel Method Diagnosis and Maintenance for Key Produce Plant Based on MAS and NN

Weijin Jiang[1] and Xiaohong Lin[2]

[1] School of computer, China University of Geosciences, Wuhan 430063, China
jwjnudt@163.com
[2] Zhuzhou Clinical Department of XiangYa Medical School , Central South University,
Zhuzhou, 412000, P.R.C.
ljxwhj@163.com

Abstract. As the development of the electrical power market, the maintenance automation has become an intrinsic need to increase the overall economic efficiency of hydropower plants. A Multi-Agent System (MAS) based model for the predictive maintenance system of hydropower plant within the framework of Intelligent Control-Maintenance-Management System (ICMMS) is proposed. All maintenance activities, form data collection through the recommendation of specific maintenance actions, are integrated into the system. In this model, the predictive maintenance system composed of four layers: Signal Collection, Data Processing, Diagnosis and Prognosis, and Maintenance Decision-Making. Using this model a prototype of predictive maintenance for hydropower plant is established. Artificial Neural-Network (NN) is successfully applied to monitor, identify and diagnosis the dynamic performance of the prototype system online.

1 Introduction

The valid method that raises the economic performance of hydropower plants is using predict maintenance system, and taking the request of the hydropower plants control, maintenance and the technocracy comprehensive into account, using the thought of the intelligence control-maintenance-management (the Intelligent Control Maintenance System, ICMMS), carries out the integration of the hydropower plants control, maintenance and the technocracy functions[1-3].

The distribution of space and logical is inherent in the water electricity production, the process, from get data to the maintenance decision, is a complicated system that is constitute by many statures processes, so the establishment and realization of support the predict system in the water power station is a kind of solution, which can solve the complicated distribution problem[4-6]. The Multi-Agent System (MAS) Theory is a kind of intelligent integration method based on the distribute type foundations, provide the new understanding angle of view and the theories frame for resolve the problem of handed over with each other under the complications, distribution environment, provide a new path to set up the model of the complicated system, to analysis, design and realize to the complicated system[7-10].

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 870 – 879, 2006.

There are many resemblances between the information change process, which happened between the agent of multi-agents under the complication[11], dynamic environment, and the issue and object[12], to which the predictive maintenance system of hydropower plant relate with the framework of ICMM[12-15], so, the predictive maintenance system of hydropower plant under the framework of ICMM also can be established by dint of the thought of agent. In the same time, the embed microprocessor abroad apply at the scene, and computer network and software technique are maturity, which make it's highly possibility that the predictive maintenance system of hydropower plant based on the thought of agent[16-18].

Under the framework of ICMMS, the predictive maintenance system of hydropower plant function is divided to four layers: Signal Collection, Data Processing, Diagnosis, and Maintenance Decision-Making, and so, according to the predictive maintenance function bed model and multi-agent theory, the multi-agent model of predictive maintenance system of hydropower plant is established[19-21]. And identify and diagnosis model based on NN is proposed, which to achieve the ante-type system of multi-agent model.

2 Multi-agent System Model of the Predictive Maintenance System of Hydropower Plant

2.1 Layer Model for the Predictive Maintenance System of Hydropower Plant

The function of the predictive maintenance system of hydropower plant included all kinds of aspect from Signal Collection to the Maintenance Decision-Making establishment. For the better comprehension to predictive maintenance and its implemented in the engineering, the predictive maintenance system can be divided into some layers, these currently function layers can be use to show an integrity of predictive maintenance system, but the contents of each layer need to confirm, according to practice apply.

Function layers of the predictive maintenance system of hydropower plant are divided into four layers: Signal Collection, Data Processing, Diagnosis, and Maintenance Decision-Making. Moreover, in order to realize the commutation between person and machine, it also can add one layer-denotation layer. Each layer has the ability of request and sends out data to any other function layer, it said that the information could spread from bottom layer to upper layer, also could spread from upper layer to bottom layer. The fluxion of the data usually takes place between draw near function layer, but in the actual application, considering the efficiency of the information processing, the data may probably cross mesosphere, transfer to a purpose layer directly. Moreover, each layer's data can transfer directly to denotation layer using to the change between person and machine.

2.2 Maintenance-Agent

The Maintenance Agent is a structure of an BDI agent, In the meantime, it has a characteristic of thing deeply and respond agent, and has a ability of affairs drive (respond to the abrupt affairs of the environment in time) and a ability of target drive (take action to the environment), under the frame work of Intelligent Control Maintenance Management System (ICMMS), establish the model of predictive maintenance system.

The maintenance agent: A maintenance agent can be figured by four buck group<Beliefs, Events, Goals, Plans>.

The maintenance agent::=<Beliefs, Events, Goals, Plans>

The Beliefs that in the formula is world knowledge, including the cognition of oneself and environment, the knowledge is an essential element of maintenance agent; Events is a variety collect which needs maintenance agent make respond to; The Goals is a target collect of maintenance agent, these targets is coming from oneself or outside environment, and may be caused by the request of other maintenance agent, the result of event or the variety of belief; Plans are the programming collect that the description the maintenance agent how to respond to the environment variety (Events) and achieve goals.

Fig. 1 is the result of sketch map that supports the maintenance agent, it is composed of sensor, knowledge database, programming, controller, desire, intent, executer and user interface.

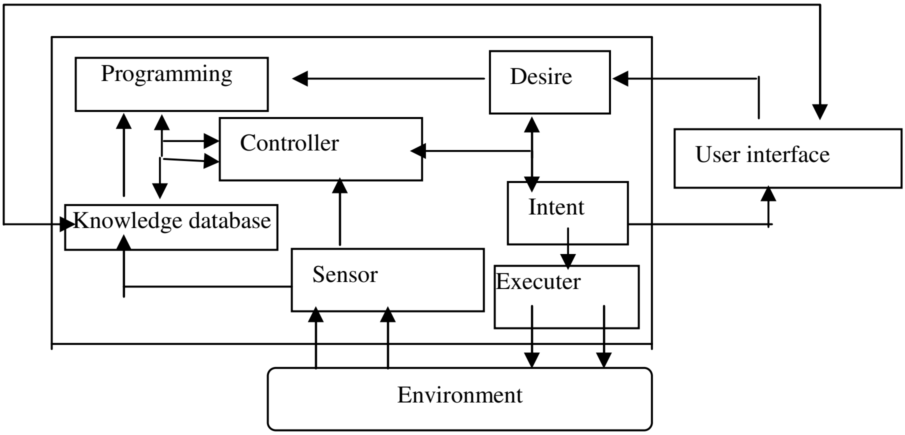


Fig. 1. Structure of Maintenance-Agent

Maintenance agents take change with world by oneself database, sense the variety of the world by sensor module, acquire information and deposit it in knowledge database, execute by executer module, achieve intention, realize target. Maintenance agents work on outside environment by information or behavior. User can query and renewal knowledge database of maintenance agent by user interface.

Maintenance agent can not only keep responding to the urgent circumstance, but also can make use of certain strategy to make a programming for the behavior of short term, then using analysis model of world and other maintenance agent build up to predict future state, and using the communication language to cooperate and consulate with other maintenance agent. Then, on the one hand, they can satisfy the request of real time and deal with the outburst hitch in time, on the other hand, they can also apply various high-level arithmetic and ratiocinate ways to make predictive maintenance system validity from overall.

2.3 Multi-agent Based Model for Predictive Maintenance of Hydropower Plant

Predictive maintenance of hydropower plant has characteristic of space and logic. It can be thought that the predictive maintenance system is a multi-agent system which is composed of several interaction maintenance agents, these maintenance agents achieve function of each layer of predictive maintenance system, and by cooperating they achieve maintenance system function.

Multi-agent based model for predictive maintenance of hydropower plant by fig 2, include Data collection agent, Data process agent, Diagnosis and Prognosis, and Maintenance Decision-Making agent. They achieve a series of function from signal collection to maintenance decision-making of predictive maintenance. Each agent has different work, the range of function also different. In addition, for predictive maintenance purpose, the cooperation agent mainly takes place between homology and border layer. The both direction arrowhead of fig 2 is mutual relation of agent, multi-agent system and production process.

Agent needs to communicate to change information, correspond or cooperate, achieve mission. The implement of Agent message depends on the concrete equipments and exploiter language of agent, for example, we can use assemble language for agent which is on line. Other layers can use C, C++ or other higher languages. This kind of message model of communication shields the detail of communication protocol of bottom, computer network technique and software bus technique(CORBA、DCOM、OPC etc.) have already made the application of different equipments, applied procedure of equipments can correspond each other.

3 Application of NN in Multi-agent Model a Prototype of Predictive Maintenance System

3.1 Prototype Model of Multi-agent Predictive Maintenance System

The terrace of ICMMS with turbo-generator as object, research the integration of the control function, maintenance function and technocracy function of timing system. In the terrace of ICMMS, multi-agent model of sub-predictive-maintenance system is an importance part; it is a prototype model of multi-agent predictive maintenance system (fig. 3). For the terrace of ICMMS, data collect agent have machine frequency measure unit、net frequency measure unit, intelligent guide leaf electricity fluid servomechanism、intelligent oar leaf electricity servomechanism, data process agent is a function module based on NN, achieve state identify and track to electricity fluid servomechanism, Diagnosis and Prognosis agent is established to electricity fluid servomechanism, accomplish health monitor, hitch diagnosis, demotion diagnosis and track function, Maintenance Decision-Making agent is according to diagnosis and prognosis agent result, combine practice circulate estate, then make maintenance decision-making. Predictive maintenance system in the ICMMS, data process agent, diagnosis and prognosis, maintenance decision-making is achieved in the same computer, data mutual though Profibus-FMS+OPC between data collection agent.

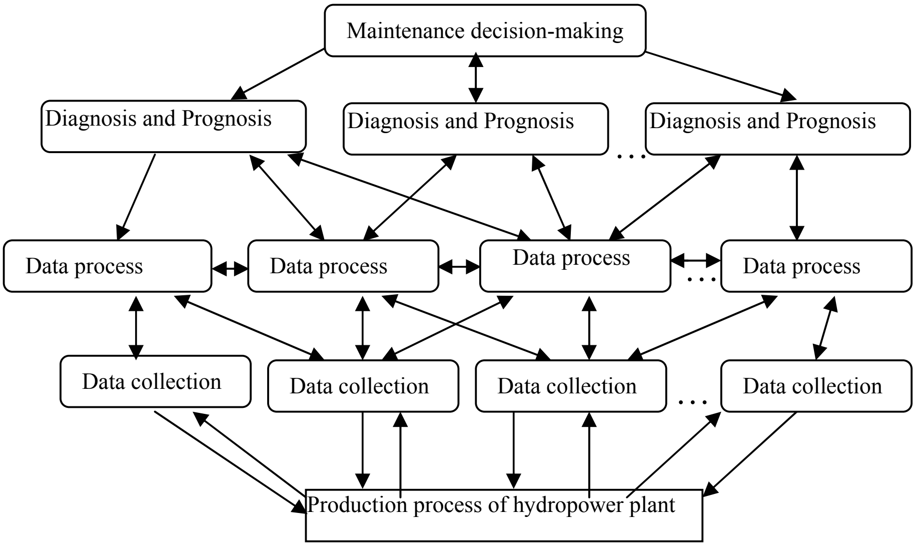


Fig. 2. Multi-agent based model for predictive maintenance of hydropower plant

3.2 Identify and Diagnosis Model Based on NN

Fig. 4 gives that we make use of serial-parallel identify model to carry out dynamic identify of electro-hydraulic servomechanism in ICMMS, at the same time, we use parallel identify model to carry out some hitch process. The adopted NN is a structure of 4*7*1 three layers feed forward NN, four input of NN is: U(K)、U(k-1)、Y(K) and Y(K-1), U is the control capacity, Y is host relay ware, K is the sample time; The output Y*(K-1) is host relay ware of NN predictive.

For obtaining initial weight of NN model, we train the NN off-line by using the acquired physical model, it is said that experiment the form.

Fig. 5 is the identification and prognosis model based on NN. To monitor electro-hydraulic servomechanism state on online, we adopt Serials-parallel identification model to carry out dynamic state of electro-hydraulic servomechanism, the history value of output of NN is actual history value of electro-hydraulic servomechanism. In the each sample time, we compare the output of NN with electro-hydraulic servomechanism; there are three possibilities:

$$\text{The } d_{if} \leqq e\,1, \quad e1=0.001\% \tag{1}$$

$$e1<d_{if}<e2, \qquad e2=0.05\% \tag{2}$$

$$\text{The } d_{if} \geq e\,2 \tag{3}$$

The d_{if}: the dispatch value of the output of NN and output of electro-hydraulic servomechanism; e1: the demotion valve value of electro-hydraulic Servomechanism; e2: the conk manages value of electro-hydraulic servomechanism.

Pattern 1 means the dynamic state consistent of NN and electro-hydraulic servomechanism, the weight of NN needn't adjust; Pattern 2 means that the dynamic state has a warp of NN and electro-hydraulic servomechanism, we should discipline to NN for running after dynamic state alter of electro-hydraulic servomechanism, so when the system has a variety of parameter evocable the variety of system characteristic, NN can reflect this change, pattern 3 means the parameter or structure happens acuity variety of electro-hydraulic servomechanism, it has a hitch occur, need adopt relevant an emergency measures.

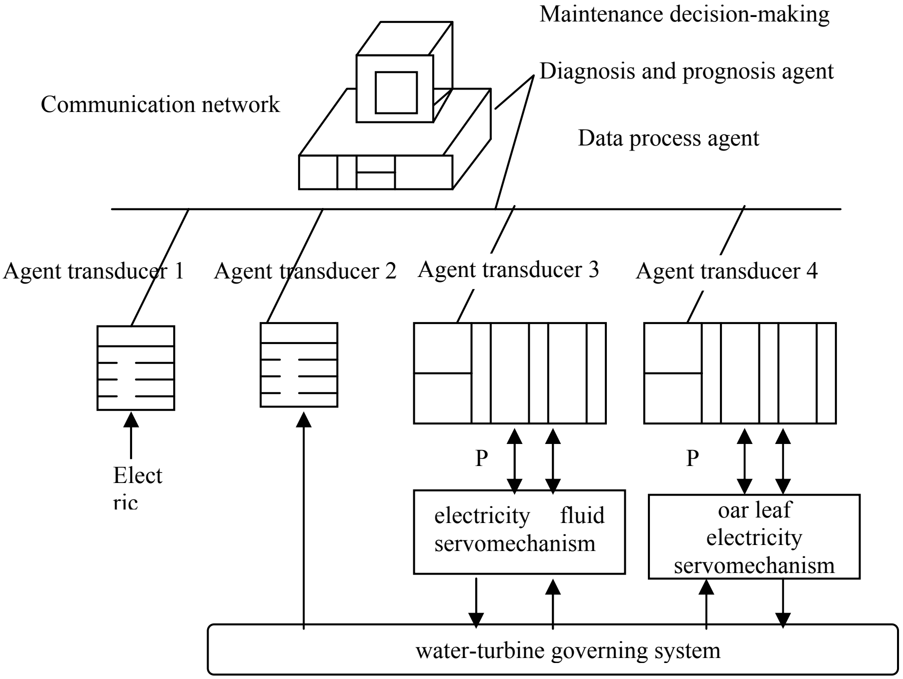

Maintenance decision-making
Diagnosis and prognosis agent
Communication network
Data process agent
Agent transducer 1
Agent transducer 2
Agent transducer 3
Agent transducer 4
Elect
ric
P
P
electricity fluid servomechanism
oar leaf electricity servomechanism
water-turbine governing system

Fig. 3. Prototype system of MAS-based model for predictive maintenance

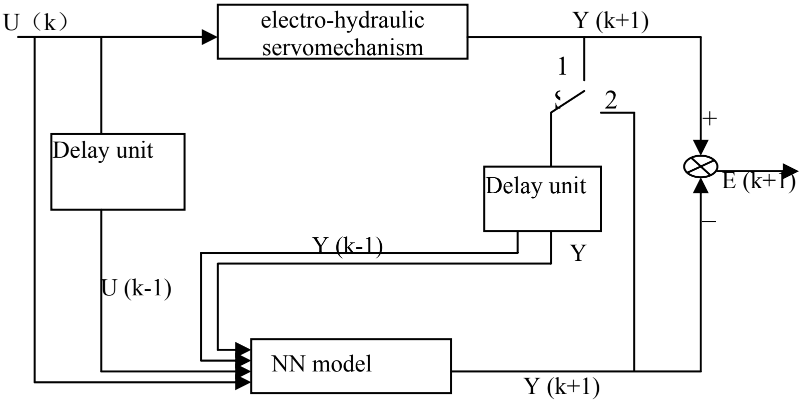


Fig. 4. Parallel/Serials-parallel identification modelbased on NN

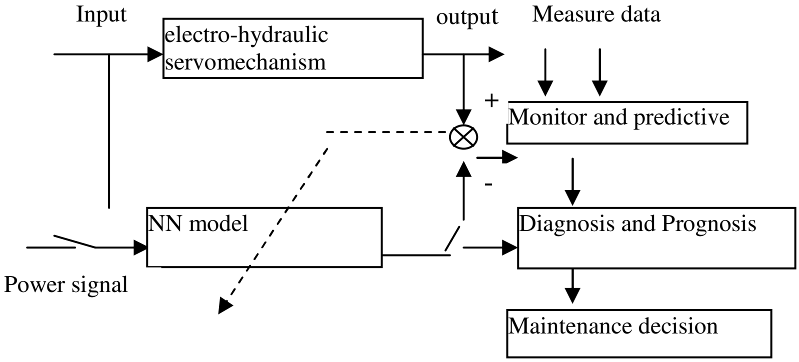


Fig. 5. Identification and prognosis model based on NN

3.3 Diagnosis and Prognosis Model Based on NN

The abnormity state of electro-hydraulic servomechanism is divided into serious hitch and demotion, they can reflect the change of inside characteristic of electro-hydraulic servomechanism, according to over discussion, and these hitch and demotion can be identified by NN.

When we choose the time of diagnosis and prognosis, we cut input signal of NN system, but add the inspirit signal of choosing to its input, so, NN adopts parallel connection identify model, it needs the history value of output adopt the history value of oneself. When we calculate the output of NN, and basic input and output, we can distill character measure that efficiency of reflect some hitch, then the value of character measure and change direction carry out diagnosis and prognosis to this hitch circs. The research surveying this way is very efficiency to electro-hydraulic servomechanism demotion process.

For example, though calculation of analysis and imitate, we discover that, when the feedback hitches, the obvious affect is warp of electro-hydraulic servomechanism. When feedback breaks off, the output of steady state rivet max is one, when feedback departs, the warp of the steady state output and the excursion extent is a function relation that is concatenation and barren, when the breadth value of input signal is 0.5, the relation between the steady state error and feedback excursion extent is fig. 6. So, we can choose the signal of step regards as special inspirit signal, the characteristic of between steady-state error and feedback coefficient of output. In the ICMM, we enact the feedback coefficient of electro-hydraulic servomechanism is 0.003/s, the 0.5 value of input is invariableness unaltered, adjust 54 hypo in the 180s of NN, it achieves track of electro-hydraulic servomechanism characteristic. More making use of multinomial exponent flatness predictive and so on, we can predictive excursion of any time in the future, according to history and value of nonce. So, we also can diagnose and prognosis to all kinds of feedback hitch and demotion of electro-hydraulic servomechanism.

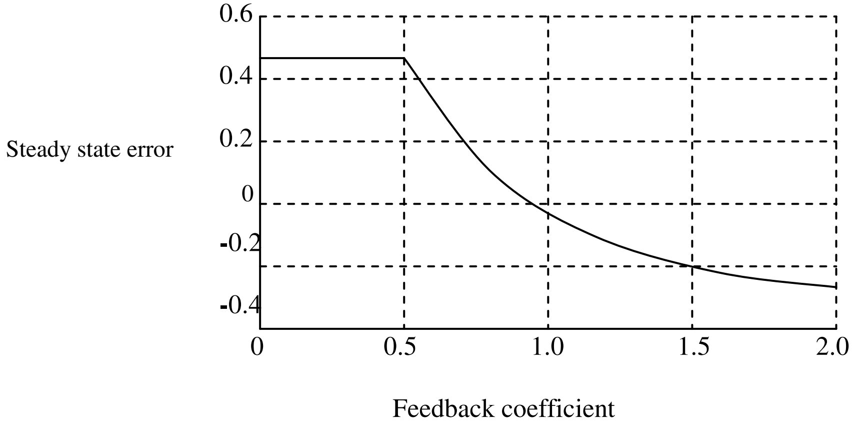


Fig. 6. Relationship between steady-state error and feedback coefficient

3.4 Maintenance Decision-Making Based on NN

If the parameter of electro-hydraulic servomechanism or the result changes acutely in the running process, the output of electro-hydraulic servomechanism and output of NN warp is satisfied formula (3), so the system adopts an emergency measures, and calls off adjusting to the parameter of NN, NN didn't track equipment state, meantime, the identification model of NN changed serials-parallel into parallel-serials, the system entry hitch insulate and maintenance state. So the preservative information of NN reflects that system can be acceptable before hitch happen, according to error carry out hitch diagnosis, until hitch eliminate, then resume the parameter adjust of NN.

4 Conclusions

(1) The result of the predictive maintenance system of hydropower plant, within the framework of Intelligent Control-Maintenance-Management System (ICMMS), being compose of data collection, data processing, Diagnosis and Prognosis, and Maintenance Decision-Making, is proposed. It has been proved that through the application instance above, on the basis of optimization neural network design method of IMSE, with combining evolved intergrowth algorithm and density of immune principle suppress regulation mechanism together, system have shortened the individual's length of code and lightened the calculating amount by solving the evolution of the colony to the neuron part. Meanwhile, system adopted the improved immune adjustment algorithm, which improved the variety of the colony effectively. The neuron that produced in the colony in this way can quickly get and realize the network, which is controlled by the thick and shape of the board.

(2)A Multi-Agent System (MAS) based mix model of the predictive maintenance system of hydropower plant within the framework of Intelligent Control-Maintenance-Management System (ICMMS) is proposed, and the multi agent model is established of hydropower plant. Using this model, a prototype of predictive maintenance for hydropower plant is established.

(3)The multi-agent model of predictive maintenance system of hydropower plant also is applied to control system in two hydropower plants strobe. The two set of hydropower plant control system have carry for more than two years.

Acknowledgments

This work is supported by the Natural Science Foundation of Hubei Province of China (No. 2001ABB006; 2003ABA043), and the Plan of new technology projects in China National Packaging Corporation 2005 (No. 05ZBJA011).

References

1. Yu Ren,Zhang Yonggang,Ye Luqing et al.The analysis and design method of maintenance system in Intelligent Control Maintenance Technical Management System (ICMMS) and its application[J].Proceedings of the CSEE,2001,21(4):60-65.
2. Chen Changzheng, Su Qing, Liu Yifang et al. Intelligent fault diagnosis inethodfor turbo-generator unit[J]. Proceedings of the CSEE, 2002, 22(5):121-124.
3. Barata J,Guedes Soares C,Marseguerra M et al.Simulation modeling of repairable multi-componse deteriorating systems for on conditon'maintenance optimization [J].Reliability Engineering and System Safety.2002,76(3):255-264
4. IEEE Task Force,Risk,and Probability Applications Subcommittee.The present status of maintenance strategies and the impact of maintenance on reliability [J].IEEE Transactio on Power System,2001,16(4):638-646.
5. Wang Shanyong,Fan Wen,Zhong Dunmei.Condition maintenance decision-making system based on condition monitoring of main equipment of hydroelectric plant[J].Automation of Electric Power System,2001,25(1):29-32.
6. Maillart L M,Pollock S M.Cost-optimal condition monitoring for predictive maintenance of 2-phase systems[J].IEEE Transactions on Reliability.2002,51(3):322-333.
7. Cao Zhongzhong,Yang Kun,Gu Yujiong et al.Quantitative RCM analyzing method for feed pump units in power plant[J].Proceedings of the CSEE, 2003,23(9):207-211.
8. Peng Hui,Zhang Yan,Zhang Yankui et al. Research on the optimized plNNed-maintenance period of turbo-generator[J]. Proceedings of the CSEE, 2003, 23(7):41-45.
9. Cheng Zhihua, Zhang Jiamguang. Application of condition based maintenance technology and its computer aided analysis system[J]. Electrical System Technology, 2003, 23(7): 41-45.
10. Liu Youguang, Li Guangfan, Gao Keli et al. Fundamental frame to draft "Guide Condition maintenance of electric power equipment"[J].Electricity system Technology, 2003,27(6):64-67.
11. Wooldridge M J,Jennings N R. Intelligent agents: Theory and practice[J].Knowledge Engineering Review,1995,10(2):115-152.
12. Vidal J M,Buhler P A,Huhns M N.Inside an agent[J].IEEE Internet Computing, 2001, 5(1): 82- 86.
13. Zhou Ming,Ren Jianwen,Li Gengyin et al. A multi-agent based dispatching operation instruction system in electric power system[J].Proceedings of the CSEE, 2004,24(4): 58-62.

14. Vitturi S. On the use of ethernet at low level of factory communication systems [J]. Computer Standards and Interfaces,2001,23(4):267-277.
15. Chen Shuyong,Li Jian, Bai Xiaomin. Research on standardized application program interface based on CORBA[J].Proceedings of the CESS, 2002,22(6):16-18.
16. Shin Jnho, Park Sungsik. CORBA-based integration framework for distributed shop floor control[J].Computers and Industrial Engineering,2003,45(3):457-474.
17. Levin U, Narenda K S. Control of nonlinear dynamical systems using networks-Part II: observation, identification, and control[J].IEEE Trans. Neural Networks,1996,7(1):33-42.
18. Jiang Weijin. Research on optimal prediction model and algorithm about chaotic time series. Journal In Miniature Microcomputer system, 2004, 25(12):2112-2115.
19. Jiang Weijin, Xu Yusheng, Sun Xingming. Research on Diagnosis Model Distributed Intelligence and Key Technique Based on MAS. Journal of Control Theory & Applications, 2004; 20(6): 231-236
20. Liu Guiquan,Chen Xiaoping,Fan Yan, *et al*. A formal model of multi-agent cooperative systems. Journal of Computer, 2001, 24(5):529-535

Nonlinear Hydrological Time Series Forecasting Based on the Relevance Vector Regression

Fang Liu[1], Jian-Zhong Zhou[1], Fang-Peng Qiu[2], Jun-Jie Yang[1], and Li Liu[1]

[1] School of Hydropower and Information Engineering,
Huazhong University of Science and Technology,
Wuhan, Hubei 430074, China
jz.zhou@mail.hust.edu.cn
[2] School of Management, Huazhong University of Science and Technology,
Wuhan, Hubei 430074, China

Abstract. As long leading-time hydrological forecast is a complex non-linear procedure, traditional methods are easy to get slow convergence and low efficiency. The basic relevance vector machine (BRVM) and the developed sequential relevance vector machine (SRVM) are employed to forecast multi-step ahead hydrological time series. The relevance vector machine is a sparse approximate Bayesian kernel method, and it provides full probabilistic forecasting results, which is helpful for hydrological engineering decision. BRVM and SRVM are respectively applied to the annual coming runoff forecast of Three Gorges hydropower station as case study. When compared with auto regression moving average models, BRVM exhibits high model efficiency and provides satisfying forecasting precision. SRVM is potential for its increased freedom and adaptive model selection mechanism. Comparison is also made within direct forecast and iterative one-step ahead forecasting for multi-step ahead forecasting, and the latter shows the ability of highlighting the model performance.

1 Introduction

Hydrological forecast is a very complex procedure with highly nonlinear characteristics in spatio-temporal changes of hydrological time series. If such forecast modeling is based on linear or approximately linear methods, the inherent limitations are inevitable. As its importance in exploring and optimizing water resources management, long leading- time stream flow forecast with high accuracy gives more scientific and efficient instructions to flood prevention, reservoir regulation and drainage basin management. Due to the complex non-linear process, such forecast is generally built on qualitative analysis, since the corresponding quantitative analysis has greater errors especially for extreme values of runoff.

Methods have been adopted to solve this problem. Statistics forecast is used most [1]. Its basic principle is to seek and analyze the change rules of hydrology ingredients and the relations with other factors by statistics. Regression is one of the most common methods in hydrological time series forecast, and has gained great improvements and broad application in other area. Although pure regression-based forecasts often achieve high skill when preforecast conditions are within the range of past

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 880 – 889, 2006.

observations, they can perform poorly in conditions outside or near the limits of the data used to estimate the regression coefficients [2] and they also have the disadvantage of amplifying frequency noise in the data when differencing. Recent researches reveal that artificial neural networks (ANNs) have been widely used for water resources variables modeling [3, 4]. As ANNs are nonlinear data-driven methods, they suit well to nonlinear input-output mapping techniques. However, there inevitably exists low convergence and local optimum problems when hydrological forecasting. Fuzzy theory have obtained more concern in hydrology for its convenient transition between natural language and mechanical inference [5], but the transition measurement is still an obstacle. Since hydrological time series are multidimensional, nonlinear, and noised, it is hypothesized to be chaotic in ref. [6] and analysed by combining the macroscopic and microcosmic spatio-temporal scales, whereas the presupposition that the series are chaotic needs deep research and discussion.

As the aforementioned content, there are many computation tools for time series regression which can predict well regular series, but the opened problem is how to design tools that have ability to model well and fast also drifting and non-stationary data [7]. In 2001, Tipping [8] explicated the concept and algorithm of relevance vector machine (RVM). RVM is a nonlinear sparse machine learning algorithm, and with Bayesian inference, it has better generalization. RVM employs the identical function form to support vector machine (SVM), but compared with SVM, RVM has the superiorities of supplying probabilistic output information, fixed hyperparameters and easy realization. Some improved RVM algorithm was presented to meliorate its application [9]. Ref. [10] describes a highly accelerated algorithm for marginal likelihood maximization in RVM, and the sequential training of RVM is presented in [7] for the purpose of perform simultaneous optimization of important parameters.

The remainder of this paper is arranged as follows. Section 2 provides a brief introduction to relevance vector machine with particular reference to the time series regression. Section 3 introduces the learning algorithms of basic relevance vector machine and the sequential relevance vector machine. In Section 4, the hydrological time series multi-step forecast of Three Gorges hydropower station using relevance vector regression is employed as case study, together with describing the data sets and considering error indices. Results are reported and discussed between different algorithms in the end of this part. Finally, conclusions and recommendations for further work are provided in Section 5.

2 Relevance Vector Regression

Ref. [11] defines the traditional nonlinear model as:

$$t = y(x) + \varepsilon \tag{1}$$

where x is a D-dimension column input vector , t is a single output, $y(\cdot)$ is a nonlinear function , and $\varepsilon \sim N(0, \sigma_\varepsilon^2)$ is additive i.i.d Gaussian noise with variance σ_ε^2. Suppose that the training data set is $D = \{(x_n, t_n)\}_{n=1}^N$, where N is the number of training samples and t_n is the real value set. The aim of regression is to find the approximate function $\hat{y}$ with the given D .

The relevance vector regression, introduced by Tipping [8], is a probabilistic sparse kernel model identical in function form to the support vector machine (SVM), and the basic prediction function form of SVM is given in [12]:

$$y(x) = \sum_{m=1}^{M} w_m \cdot K(x, x_m) + w_0 \tag{2}$$

where $\{w_m\}, m = 1 \cdots M$ are the model weights and $K(\cdot,\cdot)$ is a kernel function. Although this model is linear in the parameters, it may still be very flexible as the size of the basis set, M , may be very large.

Generally, $p(t \mid x)$ is assumed as Gaussian $N(t \mid y(x), \sigma^2)$, and the likelihood of the data set is written as [8]:

$$p(t \mid w, \sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\{-\|t - \Phi w\|^2 / (2\sigma^2)\} \tag{3}$$

where $t = (t_1, \cdots t_N), w = (w_0, \cdots, w_N)$, $\Phi_{nm} = K(x_n, x_m)$. As training such regression model with many parameters (weights), the maximum likelihood will lead to overfitting. In RVM, a Bayesian framework is employed to pursue generalization capability. A prior distribution with hyperparameters over the weights is taken to complement the likelihood function as:

$$p(w \mid \alpha) = (2\pi)^{-M/2} \prod_{m=1}^{M} \alpha_m^{1/2} \exp(-(\alpha_m w_m^2)/2) \tag{4}$$

here, M is the number of independent hyperparameters, $\alpha = \{\alpha_1, \cdots \alpha_M\}$, which individually controls the strength of the prior over its associated weight [10]. It is this prior distribution with hyperparameters that is ultimately responsible for the sparsity properties of the model [13].

As a combination a Gaussian prior and linear model within a Gaussian likelihood, the posterior is also conveniently Gaussian and can be applied analytically in [14]:

$$\begin{aligned} p(w \mid t, \alpha, \sigma^2) &\sim N(\mu, \Sigma) \\ &= (2\pi)^{-(N+1)/2} |\Sigma|^{-1/2} \exp\{-(w-\mu)^T \Sigma^{-1} (w-\mu)/2\} \end{aligned} \tag{5}$$

where the posterior covariance and mean are respectively:

$$\Sigma = (\sigma^{-2}\Phi^T\Phi + A)^{-1}, \qquad \mu = \sigma^{-2}\Sigma\Phi^T t \tag{6}$$

with $A = diag(\alpha_0, \alpha_1, \cdots, \alpha_M)$.

During the Bayesian inference over the parameters, we can see that it is crucial to calculate the values of α_i and σ^2 . Although such estimations are not in close form, the iterative optimization will be introduced hereafter in Section 3. Now, suppose that we have found the optima α_{op} and σ_{op}^2, and given a new input vector x_*, then the approximation to the predictive distribution of corresponding output t_* is [13]:

$$p(t_* \mid t) \approx \int p(t_* \mid w, \sigma_{op}^2) p(w \mid t, \alpha_{op}, \sigma_{op}^2) dw \tag{7}$$

which has the Gaussian form: $p(t_* \mid t) \sim N(\mu_*, \sigma_*^2)$, with

$$\mu_* = F\mu, \qquad \sigma_*^2 = \sigma_{op}^2 + F^T \Sigma F \tag{8}$$

where $F = [\Phi_1(x_*), \cdots, \Phi_M(x_*)]^T$. Eq. (8) shows that, the mean μ_* is the average value of t_* over the evaluated weights, and σ_*^2 illustrates the uncertainty of predictions about the optimal values of weights.

3 Relevance Vector Learning Algorithms

3.1 Problem Description

The time series forecasting problem we concern is mainly about multi-step ahead forecasting, which can be done as direct forecast or as iterative one-step ahead forecasting [15]. As direct forecast provides consecutive values with calculating the time series only once, it takes less computation but more complexities during the nonlinear mapping procedure. In iterative one-step ahead forecasting, such complexity is much lower than that in direct case, while the calculating error is increasing during each iteration, which inevitably adds more uncertainties in the forecasting. To get a delicate trade-off, we restrict this work to iterative forecasting with tuning calculation errors in algorithm. The direct forecast is also carried out in the section hereafter to obtain integrality for the case study.

Suppose that the nonlinear hydrological time series is described as $\{y_t\}_{t=1}^T$, where T is the number of observed runoff at discrete time, and $x_t = [y_{t-1}, y_{t-2}, \cdots, y_{t-l}]$ is the model input with l time delay, then the forecasting density is obtained as $p(y_{T+1} \mid x_{T+1}) \sim N(\mu(x_{T+1}), \sigma^2(x_{T+1}))$. The following relevance vector learning algorithms are employed to give the final multi-step forecasting results.

3.2 Basic Relevance Vector Learning Algorithm

The learning algorithm for approximated Bayesian inference in basic relevance vector regression model (BRVM) is given [16]:

1) Initialization: Initialize σ^2 and $\{\alpha_t\}$ defined in Eq. (3) and (4) respectively.

2) Computation: Compute weight posterior sufficient statistics μ and Σ with Eq. (6).

3) Updating strategy: Update the $\{\alpha_t\}$ and σ^2 with the equations:

$$\alpha_t^{new} = (1-\alpha_t \cdot \Sigma_{tt}) / \mu_t^2 \qquad (9)$$
$$(\sigma^2)^{new} = (\|t-\Phi\mu\|^2)/(T-\sum\nolimits_{t=1}^{T}(1-\alpha_t\Sigma_{tt}))$$

4) Convergence: Repeat *Computation* until convergence. Generally, if the maximum iteration is reached or the gradient of the outputs are less than $1.0e^{-3}$, it converges.

5) Selection: Delete weights and basis function for which $\alpha_i \geq \alpha_{\max}$, and select the other examples as relevance vectors. Here $\alpha_{\max}$ is supposed to be infinite and taken as $\alpha_{\max} = 1.0e^5$ in this work.

6) Prediction: Make predictions via new data in the time series.

The noise variance σ^2 may be a special variable in the algorithm. In order to achieve stable performance, it can be kept fixed during *Initialization* and *Updating,* and if necessary, it changes with iterations. The convergence requirements with $\alpha_{\max}$ are also very important, since they might cause overfitting in the training procedure or redundant relevance vectors when predicting with improper values.

3.3 Sequential Relevance Vector Learning Algorithm

As the run time for the basic relevance vector training algorithm scales approximately in the cube of the number of the basis functions, Tipping presented an accelerated training algorithms for sparse Bayesian models in [10] and proposed a sequential learning algorithm based on the particular strategy that the effectiveness of the marginal likelihood maximization is dependent on certain basis properties. Due to the large additional cost associated with the extra repeated re-evaluation of the basis function and the concern of dealing well with "greedy", Nikolaev [7] developed an improved sequential approach to relevance vector regression suitable for Bayesian learning from time series. In our research, Nikolaev's algorithm is employed to get direct and iterative forecasting results of nonlinear hydrological time series.

The sequential relevance vector regression model (SRVM) [7] is mainly composed of two parts, the weights regularization and hyperparametes training. By considering the newly arrived data point and optimizing two parts simultaneously, SRVM is supposed to show better performance than off-line BRVM, and the particular information will be discussed henceforward.

SRVM is learned online based on the calculation of dynamic learning rate, and at the i-th iteration, each covariance entry $\Sigma_{tt}(i)$ is accordingly regarded as a function of certain meta-prameter $p_t(i)$, which is drawn from partial derivative of the regularized log-likelihood of the posterior mean weights. For each element t in the series, the corresponding $\Phi_t(i)$ is updated with the gain of output error and the gradual adaptation of $\Sigma_{tt}(i)$. The hyperparameters $\{\alpha\}$ are obtained by the gradient-decent renewing rule. Formulas for the training process are discussed in detail in [7]. SRVM does not deal with the changing noise variance in view of the same model stability problem. With the added three constants, namely rate change constant, stabilization constant and learning rate constant in SRVM, the computation freedom is increased which can describe the forecasting model from more aspects, but along with the uncertainty problem, and we will confer on it hereafter.

4 Case Study

4.1 Study Area and Data

Hydrological forecast is the key of the cascade hydropower stations in flood control and optimal regulation, and long leading time with high precision forecasting results is the primary problem to be settled for hydrologists. As an important factor that influences the decision in hydropower station management, hydrological forecast affects the generating schedules and benefits directly. Three Gorges drainage basin is located up Yangzi River, covering four provinces, 59756 square kilometres areas. Three Gorges cascade hydropower stations are large-scale water hinging project, and act as flood prevention, generating, and navigation for the whole drainage area. The proper hydrological forecast helps the operators well prepare for the future river situations and make right decisions.

In this part, BRVM and SRVM are verified by the real data: annual runoff time series recorded in Yichang station from 1981 to 2003, as is shown in Fig,1 with solid line. Data in the time series are normalized to avoid exceeding calculation ranges according to the equation, $Q_t^{'} = (Q_t - \overline{Q}) / Q_{std}$, where Q_t and $\overline{Q}$ are the observed and mean of runoff, and Q_{std} is the standard deviation of runoff time series. One of Yichang hydrological station's functions is to measure the coming runoff for Three Gorges hydropower station. The annual runoff is time dependant and shows significant linkage in frequency domain, which leads to the nonstationary condition. Data of 1981~ 1998 are chosen to train algorithm, and the others from 1999~ 2003 for test.

4.2 Initialization and Error Indices

The initialization is crucial to the learning algorithms based on relevance vector machine, especially for the sequential one, which may lead to no convergence and obtain totally different results. For the hydrological times series, the regression delay time is $d = 3$. The width of the Gaussian kernel is equal to $r^2 = 0.8$, and the fixed noise variance is $\sigma^2 = \mathrm{var}(y) * 0.1$. These two variables are regarded as const during the whole procedures to reach stable convergence and robotic performance, while they may be changed adaptively in a sense according to potential problems and special requirements [17]. The maximum iterations and pruning threshold are set to $iter_{\max} = 50$ and $\alpha_{\max} = 1e5$ respectively. As the common settings mentioned above, the initial value for hyperparameters $\{\alpha_i\}$ are dissimilar for BRVM and SRVM. For BRVM, $\alpha(0) = ones(1, N+1) * (1/2).\text{^}2$, which produces the vector of same values for all basis at beginning. What is more, the basis function of BRVM has bias $(1,1,\cdots,1)^T$. While in SRVM $\alpha_i(0) = 1.0e^{-N} / \Phi_i(0)$, with $\Phi_i(0)$ affording the largest initial likelihood by projecting the largest normalization onto the target vector.

Take notice of the N in $\alpha_i(0)$, and here we did not mean the length of series. In fact, when the multi-step forecasting is executed by iterative calculation, N is referred to the actual training number. The exclusive parameters SRVM has resemble that in [7].

Forecasting accuracy is estimated using the following dimensionless error measures: the Mean Absolute Error (MAE) and the Coefficient of Efficiency (CE). The Standard Error of the Estimate (SE) was also adopted to furnish an indication of the spread of errors produced by a model. The three error measures are defined according to the subsequent equations:

$$MAE = (\sum_{t=1}^{T} \left| Q_t - \hat{Q}_t \right|) / T$$

$$CE = 1 - (\sum_{t=1}^{T} (Q_t - \hat{Q}_t)^2) / (\sum_{t=1}^{T} (Q_t - \overline{Q})^2) \tag{10}$$

$$SE = \sqrt{(\sum_{t=1}^{T} (E_i - \overline{E})^2) / T}$$

in which, Q_t , $\hat{Q}_t$ are the observed and forecasted runoff respectively. $\overline{Q}$ is the mean of the observed runoff. T is the number of data for modeling. For SE, E is the error index (e.g. $Q_t - \hat{Q}_t$) and $\overline{E}$ is the mean of the error.

MAE is one common measure of forecast accuracy for continuous predictands. We favored the use of MAE because of the relatively small number of forecasts in case study. CE provides some indication of how good a model is at predicting values away from the mean. In this context, CE exhibits how well the models perform when meeting either particularly low or high runoff event magnitudes. In general, a CE value of 0.9 or above suggests ‘perfect’, above 0.8 is ‘good’, and the model is ‘unsatisfactory’ if below 0.8.

4.3 Results

To further assess the presented models, we employ the traditional auto regression moving average (ARMA) model. Forecasting with long-leading time is important for economic management of hydropower stations, and whether to forecast multi-step iteratively or give direct results for multi-step is also worth investigation. In our study, the BRVM and SRVM with iterative and direct (substituted by ‘I’ and ‘D’ respectively for convenience) multi-step ahead forecasting results are shown in Fig.1. The corresponding part of ARMA is not drawn in view of the clearance of curve effect, but its error indices are listed in Tab. 1.

In Fig.1, training data are on the left of the dash line, and on the other side are the forecasting results. BRVM fits well during training, and as alluded above, SRVM seems to obtain better performance in itself, but it gets the contrary results from overview. Ref. [7] points out that SRVM is not perfect on the training data, but it tends to show very good generalization performance on unseen data for time series regression tasks. It is positively true when we apply it to this case study by adjusting the initial parameters. The forecasting results are fairly good, but the relevant fitting procedure is quite bad (not painted here). Although it does work in some cases, the additional regularization may limit the application of SRVM and reduce its reliability with increased uncertainty. Now pay attention to the iterative and direct multi-step forecasting cases, and it is obvious that the iterative one-step ahead forecasting algorithms, whether for BRVM and SRVM, have better values and depict the curve trend during the whole procedure. The most important is the similarities for iterative and direct approaches respectively. The direct multi-step forecasting results of BRVM resemble that of SRVM, and especially in the test process, they are almost overlap in the year 1999~2001. The two iterative ones receive data alike as well.

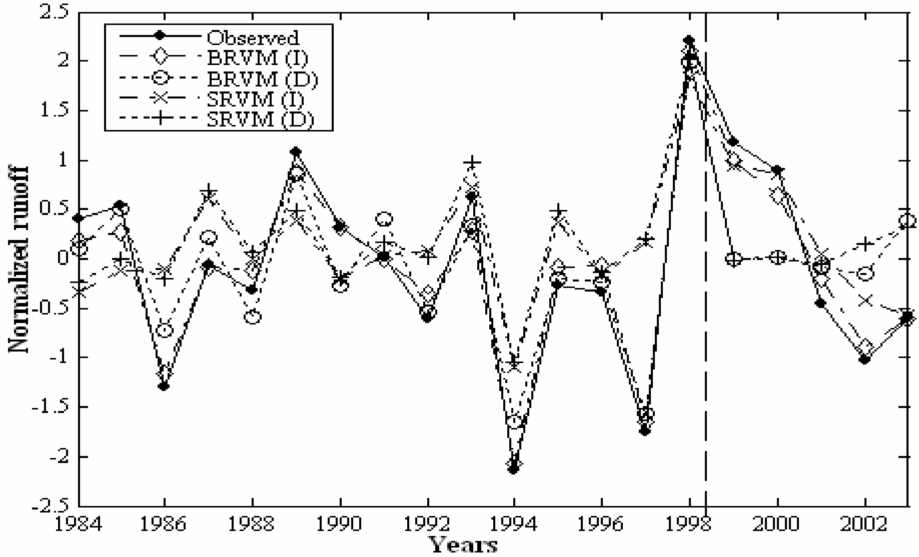


Fig. 1. Forecasting results of different algorithms for annual runoff multi-step forecasting

Error indices listed in Tab.1, including fitting and forecasting errors, are the general evaluation of presented approaches for the case study. BRVM (I) has the best values for all error indices, and with CE greater than 0.9, it is considered as a perfect model for regression, which indicates its flexibility in forecasting extreme events such as the year of very abundant runoff. On the contrary, the SRVM (D) seems to be unsatisfied with the least CE. The error indices are inquired and deemed to supplement each other quite well, which accounts for the worst of SE for SRVM (D). ARMA (I) holds the middle values of MAE and CE within their ranges, and therefore the spread of errors produced by ARMA (I) is also in the medium. Owing to the least relevance vectors, BRVM (D) learns the sparse model; however, it is not considered as a promising algorithm here with poor error indices. Although the least RVs numbers are favored, only by adopting right error indices can the forecasting model be evaluated rightly.

Table 1. Error Indices of algorithms for forecasting multi-step time series iteratively (I) and directly (D)

	MAE	CE	SE	RVs Number
ARMA (I)	0.4835	0.6521	0.5979	-
BRVM (I)	0.1594	0.9655	0.1887	10
SRVM (I)	0.5563	0.5043	0.6768	14
ARMA (D)	0.7069	0.4238	0.7524	-
BRVM (D)	0.4221	0.7238	0.5305	7
SRVM (D)	0.7218	0.3192	0.7953	17

5 Conclusion and Suggestion

The relevance vector regression learning algorithms based on sparse Bayesian model are presented and applied to hydrological time series forecasting in this study. The relevance vector machine offers essential advantages as liberal use of arbitrary kernels, no requiring estimating the error/margin parameters in advance, and the most compelling feather is that, the model is built directly within a sparse Bayesian framework with probabilistic forecast, which fits hydrological events well. The hydrological time series are stochastic, periodic and influenced by many factors. As its complexity, traditional statistics approaches are difficult to reflect its nonlinear characteristics, and BRVM is proved to be superior in our research and shows its effectiveness in convergence and efficiency to ARMA and SRVM. With regard to the discussion of multi-step forecasting problem, the iterative one-step algorithms display more reliable results than direct ones in the case study.

It should be noted that while SRVM has poor error indices, as an improvement of BRVM, it is potential for time series forecasting with the increased model freedom and good initialization. Whereas this study demonstrates the feasibility of using relevance vector regression to model hydrological time series forecasting for large hydropower station, there are still a number of areas of further work. First, it would be useful to investigate different ways of data pretreatment, such as the elimination of disturbance by denoising. Second, the delay of times series could be analyzed by imbedded dimension methods. Third, the adaptive tuning of noise variance and kernel width with persistent consistency and stability would also be expected to enhance model performance. Finally, the improved relevance vectors machine with developed initial approaches and efficient learning algorithms is preferred for its wide applications in hydrological science.

Acknowledgements

This research is supported by Nature Science Foundation of China (No. 50579022, 50539140). The support from Pre-study of National Key Basic Research Project (No. 2004CCA02500) is also appreciated.

References

1. Fan, X.Z.: Mid and Long-term Hydrologic Forecast. Hehai University Publishing Company, Nanjing (1999)
2. Guire, M.M., Wood, A.W., Hamlet, A.F., Lettenmaier, D.P.: Use of Satellite Data for Streamflow and Reservoir Storage Forecasts in the Snake River Basin. Journal of Water Resources Planning and Management 132 (2) (2006) 97-110
3. Qin, G.H., Ding, J., Li M.M., Ni, C.J.: Application of ANNs with Sensitive Ability to Hydrologic Forecast. Advances in Water Science 14 (2) (2003) 163-166
4. Yuan, J., Zhang, X.F.: Real-time Hydrological Forecasting Method of Artificial Neural Network Based on Forgetting Factor. Advances in Water Science 15 (6) (2004) 787-792
5. Liu, F., Zhou, J.Z., Yang J.J., Qiu, F.P.: The Application of Fuzzy System with Recursive Least Squares Method to Mid and Long-Term Runoff Forecast. In: Proceedings of World Water and Environmental Resources Congress, Anchorage, Alaska, USA (2005) 337-344
6. Liu. S.H., Mao, H.M.: A New Prediction of Hydrology Time Series. Engineering Journal of Wuhan University 35 (4) (2002) 53-56
7. NIkolaev, N.: Sequential Relevance Vector Machine Learning from Time Series. In: Proceedings of International Joint Conference on Neural Networks, Montreal, Canadapp (2005) 1308-1313
8. Tipping, M.E.: The Relevance Vector Machine. In: Proceedings of Advances in Neural Information Processing Systems 12, Cambridge, Mass: MIT Press (2000) 652-658
9. Shevade S.K., Sundararajian,S., Keerthi S.S.: Predictive Approaches for Sparse Model Learning. In: ICONIP 2004, Lecture Notes in Computer Science Vol. 3316 (2004) 434-439
10. Tipping, M.E., Faul, A.: Fast Marginal Likelihood Maximisation for Sparse Bayesian models. In: Proceedings of the Ninth International Workshop on Social Intelligence Statistics, Key West (2003)
11. Neal, R.M.: Bayesian Learning for Neural Networks. Lecture Notes in Statistics, Vol. 118. Springer, New York (1996)
12. Vapnik, V.N. : Statistical Learning Theory. Wiley, New York (1998)
13. Tipping, M.E.: Sparse Bayesian Learning and the Relevance Vector Machine. Journal of Machine Learning Research 1 (2) (2001) 211-244
14. Faul, A., Tipping, M.E.: Analysis of Sparse Bayesian Learning. In: Proceedings of NIPS (2001)
15. Candela, J.Q., Girard, A., Larsen, J., Rasmussen, C.E: Propagation of Uncertainty in Bayesian Kernel Models – Application to Multi-step Ahead Forecasting. In: Proceedings of ICASSP (2003)
16. Tipping, M.E.: Bayesian Inference: An Introduction to Principles and Practice in Machine Learning. In: Proceedings of Advanced Lectures on Machine Learning, Springer (2004) 41-62
17. Candela J.Q., Hansen, L.K.: Time Series Prediction Based on The Relevance Vector Machine with Adaptive Kernels. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (2003) 985-988

A Distributed Computing Service for Neural Networks and Its Application to Flood Peak Forecasting

Jun Zhu[1], Chunbo Liu[2,*], Jianhua Gong[1], Daojun Wang[1], and Tao Song[3]

[1] State Key Laboratory of Remote Sensing Science, Institute of Remote Sensing Applications,
Chinese Academy of Sciences, Beijing 100101, China
vgezj@163.com
[2] College of Environmental Sciences, Peking University, Beijing 100871, China
pklcb@263.net
[3] GIS Lab, Department of Geoinformatic Engineering, Inha University, Yonghyundong 253,
Namgu, Inchon, S.Korea, 402-751

Abstract. How to exploit current information techniques for rapidly and accurately building a fittest neural network becomes increasingly significant for flood peak forecasting. This paper firstly designs a distributed computing architecture and builds a computing environment based on Grid technologies. Then a distributed computing service for neural networks based on a genetic algorithm and a modified BP algorithm is designed and developed to rapidly and accurately building a fittest neural network for flood peak forecasting. Finally, a distributed computing prototype system is developed and implemented on a case study of the flood prevention in Shenzhen city, China. The experiment result shows that the scheme addressed in the paper is efficient and feasible.

1 Introduction

Each drainage area has its own water information monitoring networks, and these measured data could hide the gradual progress law of the river flood. Artificial neural network (ANN) is an efficient way of modeling the flood process in situations where explicit knowledge of the internal hydrologic processes is not available. In the field of water resources engineering and hydrology, ANN techniques are being used increasingly to predict and forecast water resources variables [1], and more than 43 papers have dealt with the use of ANN for the prediction of water resources variables [2]. From these research works, we can know that conventional neural networks suffered from some limitations, which may affect its application to flood forecasting. For example, the number of hidden layers and hidden neurons of the network architecture is usually determined by experiment or by trial and error; a large number of parameters are frequently required to fit a good network structure, compared to the smaller number of parameters generally required in conventional hydrological models.

In order to efficiently deal with the above situations, many experts and scholars paid attention to optimization algorithms and parallel or distributed computing

[*] Corresponding author.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 890 – 896, 2006.

techniques, and meantime implemented a lot of works [3]. The Java Object Oriented Neural Network (JOONE) is an open source project that offers a free neural network framework to create, train and test artificial neural networks. Its aim is to create a powerful environment for both enthusiastic and professional users, based on the newest Java technologies [4]. JOONE supports many features such as multithreading and distributed processing, which can take advantage of multiprocessor computers and multiple computers to distribute the processing load. Meantime its framework is also expandable with more components to easily implement new learning algorithms (i.e. new flood forecasting model). By means of JOONE technology, we can simplify much of this complexity of ANN algorithms involved in flood forecasting.

However, JOONE mainly implements its parallel or distributed computing on LAN connected by several machines, so its computing resources are limited. In addition, professionals in flood prevention, especially non-computer workers, are difficult to deploy its distributed environment. Fortunately, the development of Grid technology endows us with a promising future. Grid aims to share all the resources on the Internet to form a big, high-performance computing network. Its concept is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations. Grid Computing has a more advanced model considering resource sharing, data transfer, network security and network computing. A characteristic of Grid Computing is that it combines the merits of both parallel computing and distributed computing in network computing [5], and it provides a good mechanism of global resources sharing such as computing resources. The core of the Open Grid Services Architecture (OGSA) is the services idea, and OGSA can integrate services across distributed, heterogeneous, dynamic “virtual organizations” formed from the disparate resources within a single enterprise and/or from external resource sharing and service provider relationships [6].

The rest of this paper is organized as follows. In section 2, a modified back propagation (BP) with peak recognition theory, which can improve the accuracy of the peak value forecasting, is introduced. Moreover, we designed a global optimization model based on the genetic algorithm, which aims to avoid the network training falling into a local minimum. Section 3 describes a distributed computing environment based on Grid technology, which aims to effectively integrate more computing resources and offer more convenient and rapid computing service, which can help us rapidly and accurately obtain a fittest neural network for flood forecasting. In section 4, we designed and developed a simple prototype system, and meantime an application experiment was also implemented. Finally, we drew the conclusions with a discussion of our future research directions shown in section 5.

2 Forecasting Model

2.1 A Modified BP Algorithm

The network training in the standard BP algorithm corrects networks weight value according to the overall error, which hardly controls the training precision of the flood peak water level. In order to improve the forecasting precision of the flood peak water level, we introduce the peak value recognition theory [8] to build own flood forecasting model based on the standard BP algorithm. Compared with the standard BP algorithm correcting the network weight according to gradient descent of overall error, by this algorithm the error correction of flood peak value is mainly depending on the modification of weight according to the error of the big values. So we mainly focus on how to adapt to suitably modified coefficient of network error in training samples and modify network weight to decrease peak mapping error. We can define the network error modified coefficient ξ as $d_i^{(L)}(t)/d_{\max}^{(L)}(t)$ ($d_{\max}^{(L)}(t)$ is the biggest expectation output value of samples training). In addition, we can also modify the error's magnifying coefficient μ in order to improve the training speed of the neural network model and the precision of the peak value recognition.

2.2 Neural Network Computing Model

A key problem is that the training usually falls into a local minimum. So we must use "global optimization" techniques to explore globally the entire space of the solutions in order to find the best one network. Based on the modified BP algorithm, the genetic algorithm in this paper is used to implement global optimization. In fact, the genetic algorithm can be easily implemented by a parallel or distributed environment. We can design a network computing model and easily implement it into our distributed computing service. Figure 1 illustrates the overall process of network computing model. In this scheme, we can use the modified BP algorithm to train the neural network and the genetic algorithm to create next generation neural network for training. From figure 1 we can know that the whole process is a cycle, which will continue until at least one neural network reaches a predefined stop condition (i.e. the desired RMSE value or max cycles).

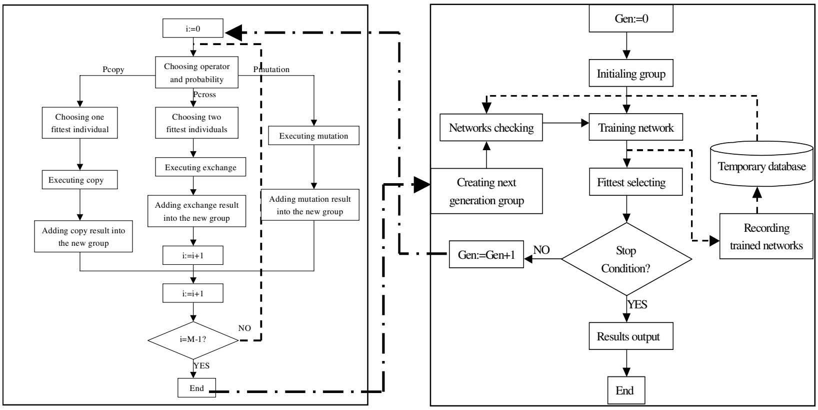

i:=0
Pcopy
Choosing operator and probability
Pmutation
Pcross
Choosing one fittest individual
Choosing two fittest individuals
Executing mutation
Executing copy
Executing exchange
Adding exchange result into the new group
Adding mutation result into the new group
Adding copy result into the new group
i:=i+1
i:=i+1
NO
i=M-1?
YES
End
Gen:=0
Initialing group
Networks checking
Training network
Temporary database
Creating next generation group
Fittest selecting
Recording trained networks
Gen:=Gen+1
NO
Stop Condition?
YES
Results output
End

Fig. 1. Flowchart of Network Model Computing

Because some individuals in next generation of the genetic algorithm are directly duplicated from the previous generation, thus the same network structure may be repeatedly trained and a lot of computing resources and time will be wasted. In order to avoid this situation, we will create a temporary database to store all trained network structure. Before training new network, we will firstly check the temporary database form and judge if the network has been trained. If the answer is yes, we will not train it and can directly get the result from the temporary database form.

3 Distributed Computing Service

In order to try different solutions and find a good neural network structure for flood peak forecast within an acceptable time, this section will focus on how to efficiently use some idle computing resources on Internet and build a neural network distributed computing service.

3.1 Distributed Computing Architecture

Based on Grid and Jini technologies, we design a distributed computing framework shown in Figure 2. Grid server manages computing resources and deals with task request. Grid services aim to effectively integrate more computing resources and offer more convenient and powerful computing service. Resources (i.e., computing resources) providers can register their resources to Grid server and these computing resources will be packed into computing resources pool. Meantime, some application services such as the neural network distributed computing service can be developed and deployed in Grid server. For an ordinary user, he/she only knows how to accesses to a given web portal for using the distributed neural network computing service. In fact, Grid mechanism can help us automatically download, install and deploy a distributed computing environment (i.e. Jini and Computefarm computing framework).

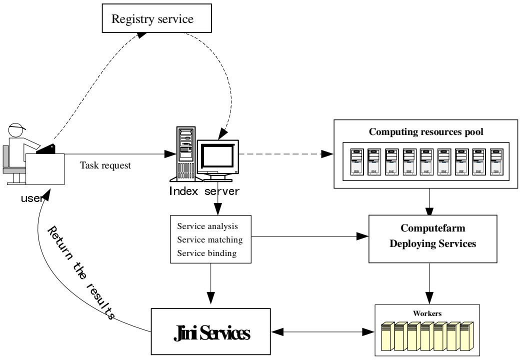


Fig. 2. Grid-Service Based Distributed Computing Mechanism

3.2 Computing Function Architecture

In this paper, the basic distributed training function framework based on Master-Worker model of Jini technology [7] is shown in Figure 3. Both the Workers and the

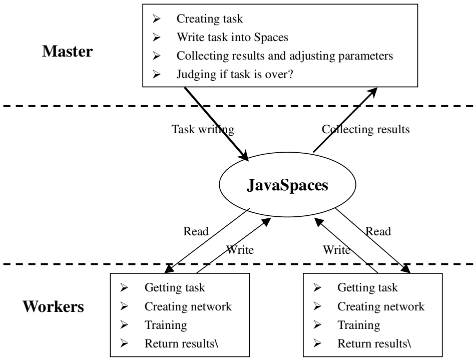


Fig. 3. Distributed Computing Function Architecture

Master use the Lookup Service to discover the JavaSpaces and the Transaction Manager services, and register themselves as listeners of the JavaSpaces in order to be notified when a neural network is available on it to be elaborated. The Master generates all the tasks and sent to the JavaSpaces. Workers are notified and take that neural network, train it, and send back the results to the JavaSpaces. When a trained network is available on the JavaSpaces, the Master is notified, so it can take that network from the JavaSpaces and store it in a temporary database.

4 Application Experiment

Our application case is the flood forecasting of Shenzhen city in Guangdong province, China. Bsed on Buji river water level station (No.5) and several upriver level stations including Nigang village water level station (No.36), Sungang brake water level station (No.7), Wenjindu water level station (No.16), we aim to build the water level forecasting model of No.5 station. Training samples data is organized by measured data from 1995 to 2004, and the datum of next year is used to testing. In addition, we considered flood spread time in these stations. We scale the training data to lie within a smaller range (0, 1) to avoid the saturation when the output approaches the limits of the transfer function, which is logistic (sigmoid) tangent.

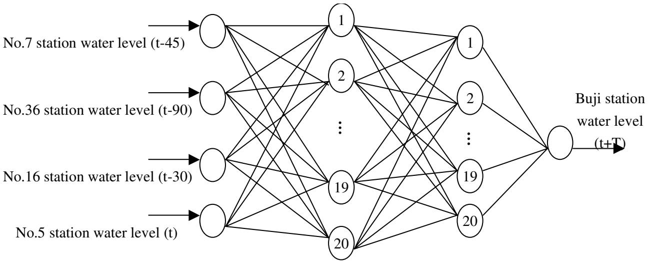

No.7 station water level (t-45)
No.36 station water level (t-90)
No.16 station water level (t-30)
No.5 station water level (t)
1
2
⋮
19
20
1
2
⋮
19
20
Buji station
water level
(t+T)

Fig. 4. Initial Network Structure of Buji Station Flood Forecasting

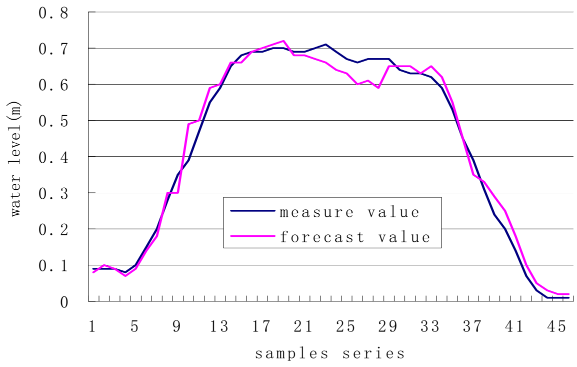


Fig. 5. The Compare of Water Level Forecasting Results with Measure Value

Based on the above scheme, we build a distributed computing prototype system using JDK 1.4, Globus toolkit 3.2, and JOONE [8], Jini2.1, Ccomputefarm 0.7. Meantime a neural network distributed computing service is implemented. We build an initial network shown in Figure 4. Some initial parameters are set as follow: $w \in (-1,1)$, μ =2.0, RMSE≤0.0001, α =0.9, η =0.0005, P=100, p_s =0.05, p_c =0.1, p_m =0.05 (α is momentum parameter value , η is learning rate). In addition, the evolution algebra is 1000, and circles of BPPR algorithm are 10000. Figure 5 shows a result of the compare water level forecasting results with real measure data and its absolute average error is less than 0.02m, which is very high precision.

Table 1 shows the time comparison of our distributed system with a stand-alone computer. From the system running effect we can see that the distributed training results are similar to stand-alone computer. In Table 1 we can see that when the network training is completed only by a node, the average response time of our system is a little longer than the stand-alone computer, because there must consume some time for data to be transferred to worker node and task dispensing. However, when nodes are added, the processing speed is accelerated and the efficiency is improved. From Table 1 we can also know that it is necessary to apply the distributed computing environment for those “global optimization” algorithms, because there are many factors that affect the processing speed of our system, such as the different input samples data, different parameters selection and so on.

Table 1. Time Comparison of Our Distributed System with Single Computer

Test conditions Performance	Single machine	Number of computing nodes				
		1	2	4	6	8
Processing time (s)	494	634	314	233	188	151
Acceleration ratio	1	0.78	1.57	2.12	2.62	3.27

5 Conclusions

Using Grid technology, this paper built a distributed computing environment. Meantime a neural network distributed computing service based on a modified BP algorithm and genetic algorithm, is also designed and deployed. All efforts aim to integrate idle computing resources and improve computing efficiency in order to more conveniently and rapidly build a fittest neural network structure for flood peak forecasting. The experiment results demonstrate our scheme can save training time of neural network and effectively forecast flood water level. In fact, there have too many complicated factors resulting in uncertainty of ANN model in the field of flood prevention especially in city. So future research efforts should be directed toward how to use or develop new techniques for deal with uncertainty for ANN model building in rapid and accurate efficiency.

Acknowledgments

This research is partially supported by the National Natural Science Foundation Project No. 40471103, the National High Technology Plan (863) Project No. 2005AA218070, and the Innovative Project of the Institute of Remote Sensing Applications, Chinese Academy of Sciences No. CX020021.

References

1. Michael, B., Yang, J.: Functional networks in real-time flood forecasting—a novel application. Advances in Water Resources, **28**(2005) 899–909
2. Holger, R.M., Graeme, C.D.: Neural networks for the prediction and forecasting of water resources variables: A review of modeling issues and applications. Environmental Modeling and Software, **15** (2000) 101–124.
3. Seiffert, U.: Artificial neural networks on massively parallel computer hardware. Neurocomputing. **28**(2004) 135-150
4. http://www.joone.org
5. Ian, F., Carl, K., Steven, T.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. (2001), http://www.globus.org/alliance/publications/papers/anatomy.pdf
6. Ian, F., Carl, K., Jeffrey M.N., Steven, T.: The physiology of the grid: An open grid services architecture for distributed system integration. Technical report, (2002), http://www.globus.org/alliance/publications/papers/ogsa.pdf
7. http://www.sun.com/software/jini/
8. Li, H., Liu, H.: Peak value recognition theory of artificial neural network and its application to flood forecasting. SHUILI XUEBAO, **6**(2002) 15–20

Automatic Inference of Cabinet Approval Ratings by Information-Theoretic Competitive Learning

Ryotaro Kamimura and Fumihiko Yoshida

[1] Information Science Laboratory
ryo@cc.u-tokai.ac.jp
[2] Department of Media Studies
Tokai University
1117 Kitakaname Hiratsuka Kanagawa 259-1292, Japan
bun@f07.itscom.net

Abstract. In this paper, we demonstrate that cabinet approval ratings can automatically be inferred with good performance by a neural network technique, that is, information-theoretic competitive learning. Because cabinet approval rating estimation is an extremely complex process with much non-linearity, neural networks may give much better performance than conventional statistical methods. Though an attempt to infer public opinions seem to be a challenging topic for machine learning, little attempts have been made to infer approval ratings to our best knowledge. In this context, we try to apply information-theoretic competitive learning to the problem of cabinet approval ratings. Information-theoretic competitive learning has been developed so as to simulate competitive processes of neurons. One of the main characteristics of the method is that it is a very soft-type of competitive learning in which conventional competitive learning is only a special case. Though the method seems to be promising due to its general property, we have had a few experimental results to show better performance. Experimental results show that without any teacher information neural networks can appropriately infer the rise and fall of approval ratings through a process of information maximization. This experiment result surely opens up new perspectives for neural networks as well as mass communication studies.

1 Introduction

In this paper, we try to estimate cabinet approval ratings by information-theoretic learning. In the field of mass communication study, there are a sizable number of studies that predict various types of public opinions from computer-generated data sets on mass media reports [1], [2], [3], [4], [5], [6]. Partly because of an absence of appropriate software for analyzing Japanese until recently, however, there have not been such studies in Japan with an exception of [7]. In addition, little attempts have been made to use machine learning techniques in the mass communication study. Because data in the study seem to be extremely complex with a property of non-linearity, machine-learning techniques,

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 897–908, 2006.

in particular, neural networks are expected to be successfully applied to the mass communication study.

In this context, we introduce information-theoretic learning, because its generalized property may improve substantially basic performance. For information-theoretic approach, there have been many attempts to use information-theoretic methods in neural networks [8], [9], [10], [11]. We have so far found similarity between competition and information maximization and proposed a new information theoretic method for competitive learning [12], [13], [14], [15], [16], [17], [18]. The new approach is a soft-type competitive learning and can solve the serious problem of dead neurons in conventional competitive learning [19], [20], [21], [22], [23], [24], [25].

In this paper, we apply the method to the inference of approval ratings of Japan's Koizumi cabinet by examining newspaper editorials. We have intuitively known that the editorials of newspapers have much influence on public opinions, in this case, the cabinet support rating. However, little formal attempts have been made to clarify this relation. Thus, we try to infer the rise and fall of cabinet support ratings by examining the editorials of Japanese newspapers. Experimental results discussed in this paper show a good potentiality of neural networks for automatic inference of public opinion.

2 Theory and Computational Methods

We have defined information content as mutual information between input patterns and competitive units [16]. As shown in Figure 1, a network is composed of input units x_k^s and competitive units v_j^s. We used as the output function the inverse of the Euclidean distance between connections weights and input patterns. Thus, an output from the jth competitive unit can be computed by

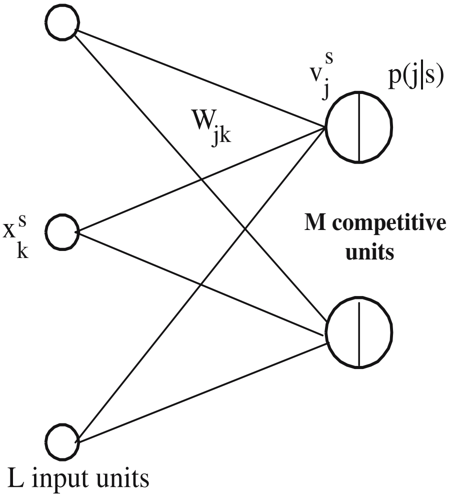


Fig. 1. A network architecture for information maximization

$$v_j^s = \frac{1}{\sum_{k=1}^{L}(x_k^s - w_{jk})^2}, \tag{1}$$

where L is the number of input units, and w_{jk} denote connections from the kth input unit to the jth competitive unit. The output is increased as connection weights are closer to input patterns.

The conditional probability of firing of the jth unit, given the sth input pattern $p(j \mid s)$ is computed by

$$p(j \mid s) = \frac{v_j^s}{\sum_{m=1}^{M} v_m^s}, \tag{2}$$

where M denotes the number of competitive units. Since input patterns are supposed to be uniformly given to networks, the probability of the jth competitive unit is computed by

$$p(j) = \frac{1}{S}\sum_{s=1}^{S} p(j \mid s). \tag{3}$$

By using these probabilities, information I is computed by

$$I = -\sum_{j=1}^{M} p(j)\log p(j) + \frac{1}{S}\sum_{s=1}^{S}\sum_{j=1}^{M} p(j \mid s)\log p(j \mid s), \tag{4}$$

where S is the number of input patterns. Differentiating information with respect to input-competitive connections w_{jk}, we have final update rules to increase information ([16]).

3 Results and Discussion

In the first experiment, we try to show that a process of information maximization accompanies a process of competition and that an artificial data can appropriately be classified into two groups as a result of competition. The artificial data was composed of patterns drawn from two normal distributions with two different means as shown in Figure 2(f). The number of input and competitive units are two, respectively. Figure 2(a) shows information as a function of the number of epochs by competitive learning and information-theoretic competitive learning. Information by information-theoretic and competitive learning are increased rapidly and reaches final stable points with about 50 epochs. No significant difference between competitive learning and information-theoretic learning can be seen in this case. Figure 2(b) to (f) show that connection weights represented in small circles are gradually expanded and located finally in the middles of two clusters. This result shows that information maximization can realize competitive processes not by the winner-take-all algorithm of conventional competitive learning but by a process of information maximization.

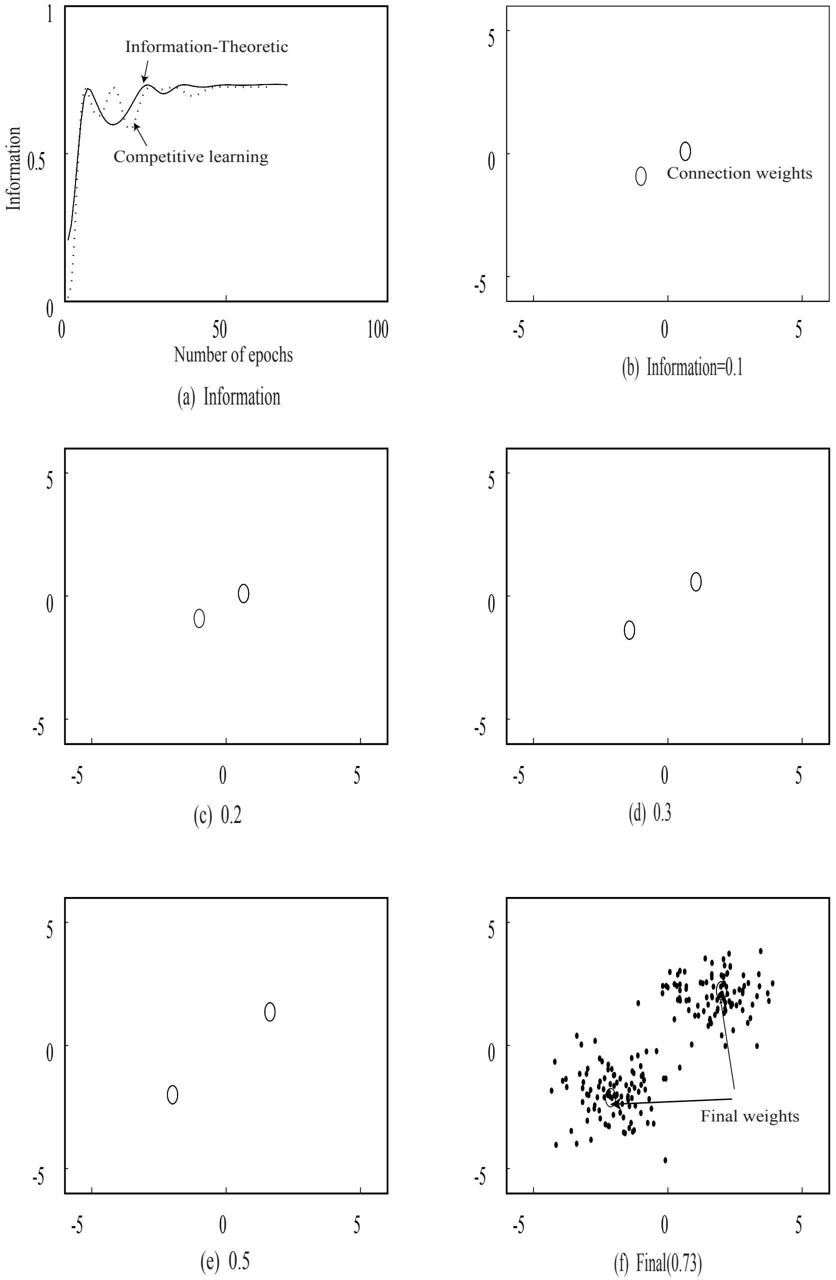


Fig. 2. Information and connection weights for five different values of information content. In Figure (a), a solid and dotted line represent information by information-theoretic and simple competitive learning.

In the second place, we apply the method to the inference of cabinet approval ratings. Most major mass-media companies in Japan independently and periodically conduct public opinion polls asking whether or not to approve the incumbent cabinet. Thus, there exist ten or more time-series data sets on approval ratings for the incumbent Koizumi cabinet. After considering consistency of survey method, frequency of polls, and availability, the one conducted by Asahi Shimbun, a major Japanese paper, was selected for this study. During the period from May 26, 2001 through September 27, 2004, Asahi conducted forty-six opinion polls basically with the once-a-month pace. On certain unexpected occasions such as an abrupt resignation of a highly popular cabinet member, however, Asahi conducted "an emergency poll" even shortly after its previous regular poll. It should be noted, thus, that intervals between two polls are not equal.

In this study, cabinet approval ratings were estimated using a data set generated by a computer software TeX-Ray from 2371 newspaper editorials of four major newspapers – Asahi Shimbun, Mainichi Shimbun, Nihon Keizai Shimbun, and Yomiuri Shimbun. These editorials cover the period through April 27, 2001, the day of inauguration of the Koizumi cabinet, through September 26, 2004, and include at least one sentence that refer to the Koizumi cabinet. From these editorials 8585 sentences that referred to the Koizumi cabinet were extracted and subsequently analyzed by TeX-Ray.

TeX-Ray, which was developed by the second author, is a computer software for analyzing Japanese sentences. It performs the following analyses for each sentence: (1) morphological analysis, (2) syntax analysis, (3) concept usage analysis, (4) positive and negative words recognition, (5) modality recognition, (6) actor-action-target triplet extraction. This study mainly used the first, fourth, and the fifth functions of TeX-Ray. Employing multiple-regression analysis, Yoshida(2006) shows that a TeX-Ray-generated data set successfully postdicted support ratings for Koizumi cabinet with exceptionally high accuracy. In this sense, it is quite reasonable to assume that the TeX-Ray-generated data set is reliable as well as valid.

Using TeX-Ray's modality recognition function and its word count function on a good-bad scale, each sentence was assessed in terms of forty variables. Of these variables, two of them assess number of positive words and negative words appearing within the last two phrases of each sentence. Here, "positive word" means a Japanese word which with no doubt most Japanese speakers regard as a word with "good" connotation. "Negative word" means, of course, the one most Japanese speakers regard as a word with "bad" connotation.

The remaining thirty-eight variables assess modality pattern of each sentence, with each variable corresponding to one of thirty-eight modality patterns. Since TeX-Ray only examines modality pattern of the last phrase of a Japanese sentence, only one matching patter is found in one sentence, at best. If a certain modality pattern is recognized, then one variable corresponding to that pattern is given a value 1, and the remaining thirty-seven variables are given a value 0.

The approval ratings for the Koizumi cabinet consist of forty-six data points. Accordingly, the data set of editorial contents was aggregated in each of forty-six periods beginning at the date of a poll and ending at the previous day of the next poll. The first period was set to start at the inauguration day and to end at the previous day of the first poll. Since the length (days) of the forty-six periods are not even, the value of each variable in each time period was divided by the intervals in that period so as to make it comparable with one in other time periods. With this data transformation, a variable belonging to thirty-eight variables that correspond to each of thirty-eight modality patterns measures an average daily frequency of a given modality pattern during a given time period.

The two variables which measure frequency of positive and negative words in a given sentence were also aggregated in each time period and were divided by the number of days in each period, thereby, transforming them into an average daily frequency of positive as well as negative words in a given time period. In each time period, these two variables were further transformed into two indices, by dividing them respectively by the average daily frequency of positive words during the entire time periods as well as by the average daily frequency of negative words during the same all periods. As a result, if in a given time period average daily frequency of positive words is greater than that of the entire periods, the value of this index is greater than 1.0, and if it is less than that of the entire periods, the value of this index is less than 1.0. By the same way, a daily frequency index for negative words was also created. Further analysis was performed using the data set of the above-mentioned opinion polls and the thirty-eight variables on modality patterns as well as two indices regarding daily frequencies of positive and negative words.

For estimating cabinet support ratings, it is necessary to match aggregated and transformed data on editorial contents in a given time period with a cabinet approval rating at a given time period. In this study, they were matched by the following way. That is, throughout the entire periods, all variables concerning with editorial contents in a given period were matched with a cabinet support rating that was surveyed at the beginning of the next time period. In other words, a cabinet approval rating at the beginning of a given time period was estimated by values of variables on editorial contents assessed during the precedent time period.

Using the above-mentioned data, we tried to infer cabinet approval ratings by neural networks. The data was so complex that we did not obtain good performance by the original data. To solve this problem, we tried to reduce the complexity of the input data as much as possible. In the first place, we used only modality variables (38 variables) as the first approximation. Among them, three variables have all zero values to be deleted in the new data seta. In addition, the last three periods have given instability in learning, and we deleted them in the new data set. Thus, we have just 36 variables (35 modality variables and a previous rating) with only 43 input patterns. Figure 3 shows two architectures for the problem. In Figure 3(a), all thirty-six variables are given into the network. Because instability in learning occurred several times, we reduced the number of variables by using the principal component analysis (minimum fraction variance component to keep=0.02). Figure 3(b) shows a situation where thirty-six variables are reduced to 15 by the principal component analysis. By these reduced variables, relatively good performance in terms of training and generalization errors could be obtained.

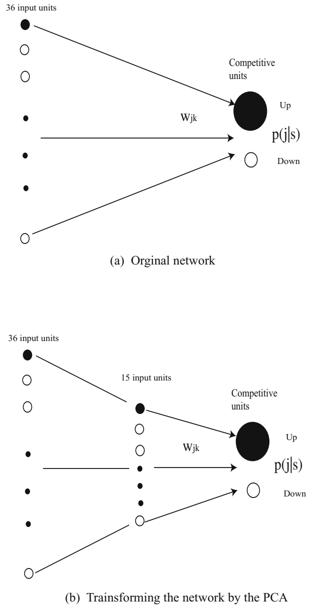
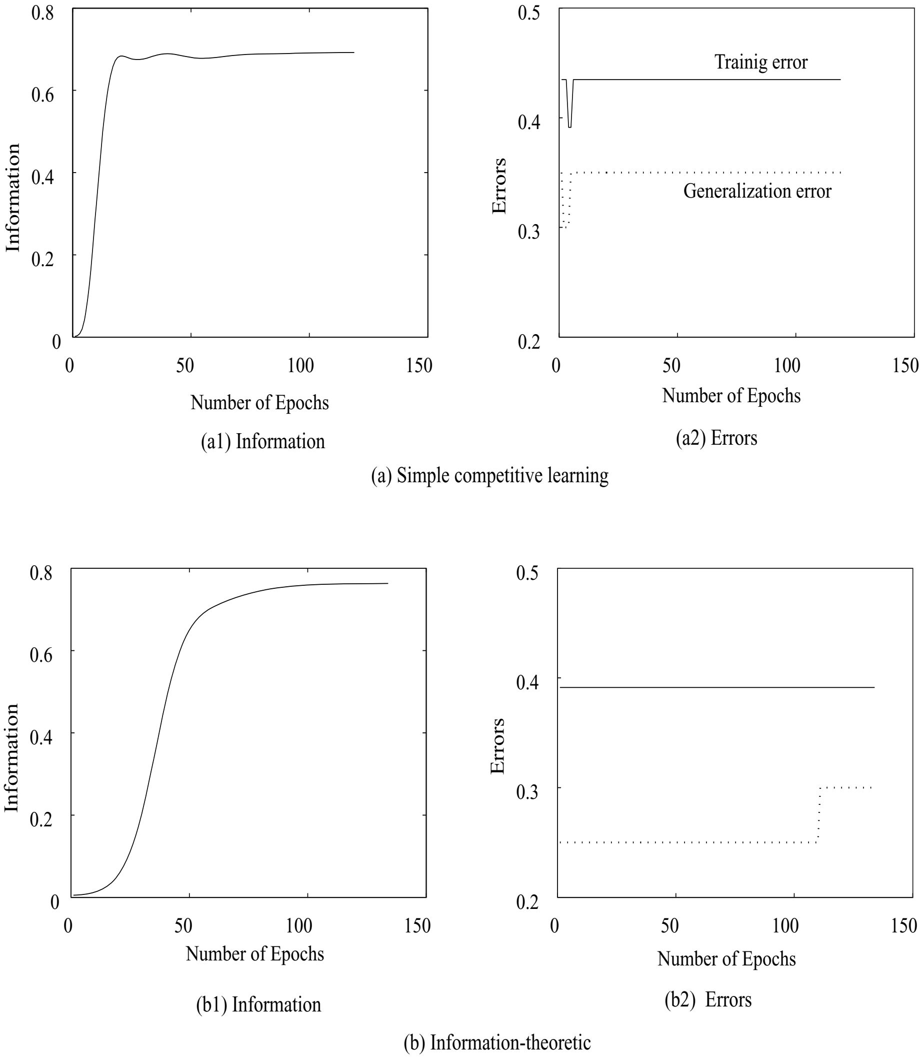


Fig. 3. A network architecture for the approval-rating problem. Figure (a) and (b) show an network architecture for the original and reduced data.

Figure 4(a) and (b) show information and errors by competitive learning and information-theoretic learning with original thirty-six variables. As shown in Figure 4(a1), information is rapidly increased to a stable point with just fifty epochs. However, Figure 4(a2) shows training and generalization errors as a function of the number of epochs. Both errors are decreased, and then increased to relatively large levels. Figure 4(b1) shows information as a function of the number of epochs by information-theoretic learning. Information increases more slowly but more smoothly to a stable point. However, as shown in Figure 4(b2), training errors are almost flat and generalization errors inversely increase at the end. Thus, it seems to be impossible to infer approval ratings by the neural networks, because the problem seems to be too complex for the networks.

We thought that the impossibility of the inference was due to redundant and unnecessary information contained in the data. Thus, we tried to condense information in input patterns as much as possible by using the principal component



(a1) Information

(a2) Errors

(a) Simple competitive learning

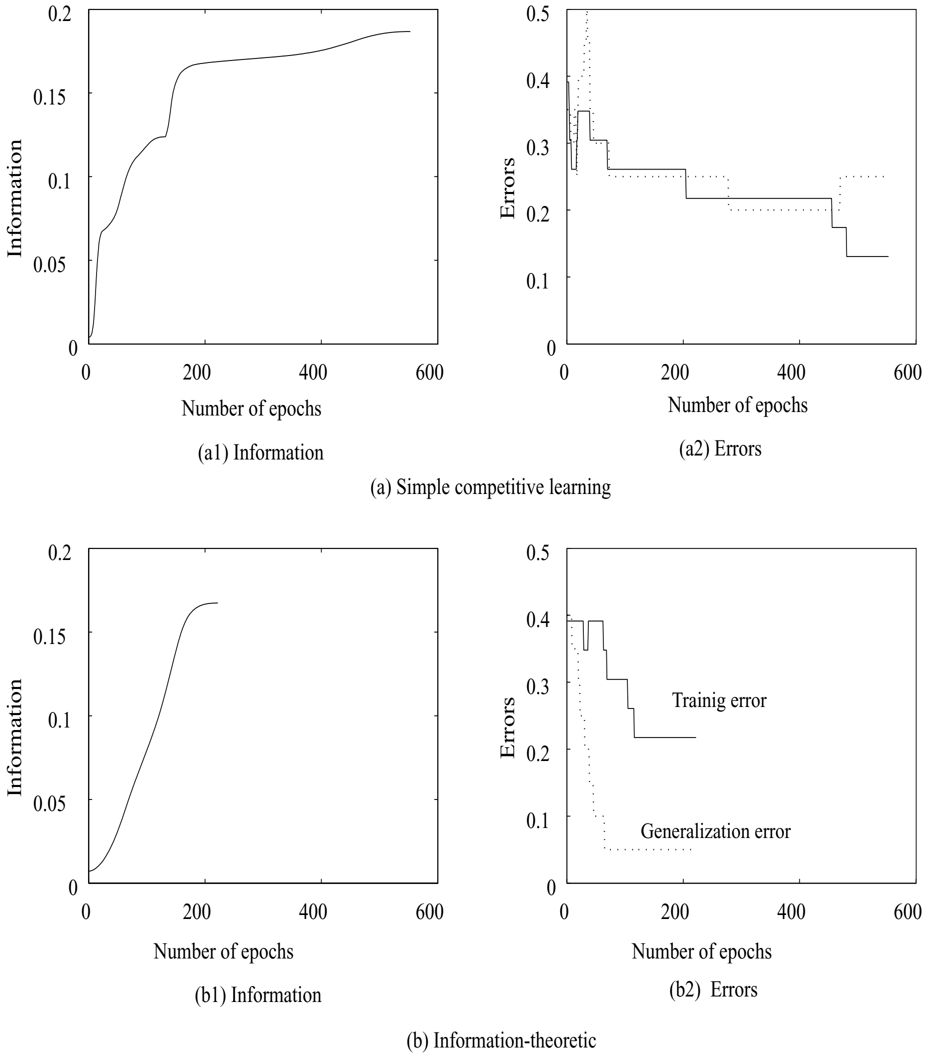
(b1) Information

(b2) Errors

(b) Information-theoretic

Fig. 4. Information and errors as a function of the number of epochs for the approval rating problem with original data. Figure (a) and (b) show results by simple competitive learning and the information-theoretic method.

analysis. By experiments, we could reduce the number of variables from thirty-six to fifteen variables. Figure 5(a) shows information and errors by simple competitive learning. Information increases with some fluctuations and approches a level of 0.2, which is lower than the level obtained by the previous model (Figure 4). Figure 5(b1) shows information as a function of the number of epochs by information-theoretic learning. As shown in the figure, information increases much more rapidly to a stable point than by simple competitive learning. Figure 5(b2) shows training (solid) and generalization (dotted) errors by information-



(a1) Information

(a2) Errors

(a) Simple competitive learning

(b1) Information

(b2) Errors

(b) Information-theoretic

Fig. 5. Information and errors as a function of the number of epochs for the approval rating problem. Figure (a) and (b) shows results by simple competitive learning and the information-theoretic method.

theoretic learning. As the training error is decreased, the generalization error is more rapidly decreased. Table 1 shows generalization comparison by three methods. We repeated experiments ten times with different initial conditions, and averaged the results for all the methods. By the conventional k-means, we had the worst performance of 0.2775. By using simple competitive learning, errors are slightly decreased to 0.2650. Finally, by using information-theoretic learning, the best result of 0.1600 could be obtained. These results show that

Table 1. Comparison of generalization errors by three methods. In the table, CL, ITCL denote standard competitive learning and information-theoretic competitive learning.

	k-means	CL	ITCL
Average	0.2775	0.2650	0.1600
Std Dev	0.1186	0.0503	0.0944

though careful preprocessing is needed, we have a high possibility that better generalization can be obtained by information-theoretic learning.

4 Conclusion

In this paper, we have demonstrated that better performance in terms of generalization can be obtained by information-theoretic competitive learning for the complex problem of the approval rating estimation of Japan's Koizumi cabinet. Information-theoretic competitive learning has been developed so as to simulate competitive processes of neurons. As information in competitive neurons is increased, a smaller number of neurons tend to be activated. When information is completely maximized, winner-take-all processes can be realized. Thus, information-theoretic learning is a very soft-type of competitive learning in which conventional competitive learning is only a special case. Though the method seems to be promising due to the general property of the method, we have had a few experimental results to show the better performance. We have applied our method to cabinet approval ratings. Because the problem is so complex and the number of variable is so large, careful consideration on the property of variables is needed. However, the experimental results in this paper have certainly shown that information-theoretic learning can be applied with better performance to actual complex problems such as the approval rating estimation. For more practical applications, we need to explore more exactly how and why better generalization performance can be improved by information-theoretic learning. However, experimental results shown in this paper certainly open up new perspectives for neural computing as well as mass communication studies.

References

1. M. M. Miller and B. Denham, "Horserace, issue coverage in prestige newspapers during 1988, 1992 elections," *Newspaper Research Journal*, vol. 15, no. 4, pp. 20–28, 1994.
2. M. M. Miller, J. L. Andsager, and B. P. Riechert, "Framing the candidates in presidential primaries: issues and images in press releases and news coverage," *Journalism and Mass communication quarterly*, vol. 75, no. 2, pp. 312–324, 1998.
3. D. Domke, D. P. Fan, S. Michael, D. V. S. Smith, and M. D. Watts, "News media, candidates and issues, and public opinion in the 1996 presidential campaign," *Journalism and Mass Communication Quaterly*, vol. 74, no. 4, pp. 718–737, 1996.

4. D. Fan, "Computer content analysis of press coverage and prediction of public opinion for the 1995 sovereignty referendum in quebec," *Journalism and Mass Communication Quaterly*, vol. 74, no. 4, pp. 351–366, 1996.
5. M. D. Watts, D. Domke, D. V. Shah, and D. P. Fan, "Elite cues and media bias in presidential campaigns," *Communication Research*, vol. 26, no. 2, pp. 144–175, 1999.
6. J. A. Danowski and A. Rebecca, *Linking gender language in news about presidential candidates to gender gaps in polls: a time-series analysis of the 1996 campaign.* Westport: Ablex Publishing, 1996.
7. F. Yoshida, "Main features of tex-ray, a software for analyzing japanese sentences, and its applications: an attempt to predict poll support ratings for koizumi cabinet from editorial content of four major newspapers (in japanese)," *Journal of mass communication studies*, vol. 68, pp. 80–96, 2006.
8. R. Linsker, "How to generate ordered maps by maximizing the mutual information between input and output," *Neural Computation*, vol. 1, pp. 402–411, 1989.
9. J. J. Atick and A. N. Redlich, "Toward a theory of early visual processing," *Neural Computation*, vol. 2, pp. 308–320, 1990.
10. S. Becker, "Mutual information maximization: models of cortical self-organization," *Network: Computation in Neural Systems*, vol. 7, pp. 7–31, 1996.
11. S. Becker and G. E. Hinton, "Learning mixture models of spatial coherence," *Neural Computation*, vol. 5, pp. 267–277, 1993.
12. R. Kamimura, T. Kamimura, and T. R. Shultz, "Information theoretic competitive learning and linguistic rule acquistion," *Transactions of the Japanese Society for Artificial Intelligence*, vol. 16, no. 2, pp. 287–298, 2001.
13. R. Kamimura, T. Kamimura, and O. Uchida, "Flexible feature discovery and structural information," *Connection Science*, vol. 13, no. 4, pp. 323–347, 2001.
14. R. Kamimura, "Information theoretic competitive learning in self-adaptive multi-layered networks," *Connection Science*, vol. 13, no. 4, pp. 323–347, 2003.
15. R. Kamimura, "Teacher-directed learning: information-theoretic competitive learning in supervised multi-layered networks," *Connection Science*, vol. 15, pp. 117–140, 2003.
16. R. Kamimura, "Information-theoretic competitive learning with inverse euclidean distance," *Neural Processing Letters*, vol. 18, pp. 163–184, 2003.
17. R. Kamimura, "Unifying cost and information in information-theoretic competitive learning," *Neural Networks*, vol. 18, pp. 711–718, 2006.
18. R. Kamimura, "Improving information-theoretic competitive learning by accentuated information maximization," *International Journal of General Systems*, vol. 34, no. 3, pp. 219–233, 2006.
19. D. E. Rumelhart and D. Zipser, "Feature discovery by competitive learning," in *Parallel Distributed Processing* (D. E. Rumelhart and G. E. H. et al., eds.), vol. 1, pp. 151–193, Cambridge: MIT Press, 1986.
20. S. Grossberg, "Competitive learning: from interactive activation to adaptive resonance," *Cognitive Science*, vol. 11, pp. 23–63, 1987.
21. D. DeSieno, "Adding a conscience to competitive learning," in *Proceedings of IEEE International Conference on Neural Networks*, (San Diego), pp. 117–124, IEEE, 1988.
22. S. C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton, "Competitive learning algorithms for vector quantization," *Neural Networks*, vol. 3, pp. 277–290, 1990.

23. L. Xu, "Rival penalized competitive learning for clustering analysis, RBF net, and curve detection," *IEEE Transaction on Neural Networks*, vol. 4, no. 4, pp. 636–649, 1993.
24. A. Luk and S. Lien, "Properties of the generalized lotto-type competitive learning," in *Proceedings of International conference on neural information processing*, (San Mateo: CA), pp. 1180–1185, Morgan Kaufmann Publishers, 2000.
25. M. M. V. Hulle, "The formation of topographic maps that maximize the average mutual information of the output responses to noiseless input signals," *Neural Computation*, vol. 9, no. 3, pp. 595–606, 1997.

Radial Basis Function Neural Networks to Foresee Aftershocks in Seismic Sequences Related to Large Earthquakes

Vincenzo Barrile[1], Matteo Cacciola[1], Sebastiano D'Amico[2], Antonino Greco[1], Francesco Carlo Morabito[1], and Francesco Parrillo[3]

[1] University "Mediterranea" of Reggio Calabria, Faculty of Engineering, Department of Informatics, Mathematics, Electronics and Transportation (DIMET),
89100 Reggio Calabria, Italy
barrile@ing.unirc.it, {matteo.cacciola, antonino.greco, morabito}@unirc.it
http://www.ing.unirc.it

[2] Istituto Nazionale di Geofisica e Vulcanologia, 00143 Rome, Italy
damico@ingv.it
http://www.ingv.it

[3] University of Messina, Department of Earth Science
98166 Messina-Sant'Agata, Italy
francescoparrillo@yahoo.it
http://www.unime.it

Abstract. Radial Basis Function Neural Network are known in scientific literature for their abilities in function approximation. Above all, this particular kind of Artificial Neural Network is applied to time series forecasting in non-linear problems, where estimation of future samples starting from already detected quantities is very hardly. In this paper Radial Basis Function Neural Network was implemented in order to predict the trend of $n(t)$ for aftershocks temporal series, that is the numerical series of daily-earthquake's number occurred after a great earthquake with magnitude $M > 7.0$ Richter. In particular we implemented the RBF-NN for the Colfiorito seismic sequence. The seismic sequences considered in this work are obtained following criteria already known in scientific literature [1], [2]. Results of proposed approach are very encouraging.

1 Introduction

Earthquakes tend to involve in cluster [3]. After the occurrence of an earthquake it is possible to observe many other ones in its proximity. In fact, about a third of them, detected all over the world, are aftershocks, the distinguishing feature of which is clustering in space and time. Their temporal distribution often follows a regular trend, as first observed by Omori in 1894. In a time series of seismic events, a mainshock is defined as an earthquake marked by greatest magnitude, while it can be possible to consider aftershocks all the earthquakes happened after a mainshock, located in a certain time interval and in a certain space from the

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 909–916, 2006.

occurrence of the first one [4], [5], [6]. In this paper Radial Basis Function Neural Network (RBF-NN) has been implemented in order to predict the trend of $n(t)$ for aftershocks temporal series, that is the numerical series of daily-earthquake's number occurred after a great earthquake with magnitude $M > 7.0$ Richter. In particular RBF-NN has been trained on California, USA (October 10, 1999) and Taiwan (September 09, 1999) sequences, and has been evaluated using the Colfiorito, ITA (September 26, 1997), seismic sequence. Structure of paper is described as follows: in section 2 the temporal rate decay of seismic aftershocks is described; section 3 depicts a theory overview about RBF-NN; section 4 describes the implementation of data set and finally in section 5 conclusions and perspectives are drawn up.

2 About the Temporal Decay Rate

The employment of statistics is useful to estimate the probability of future earthquakes. These probabilities are very interesting from the point of view of earthquake physics, and crucial for attempts to forecast the hazards due to large, damaging earthquakes. Actually a theoretical model that successfully describes earthquake recurrence is unknown, so it is necessary to adapt probability distribution based on earthquake history. Aftershocks typically occur immediately after a mainshock and are distributed through the source volume. Usually the frequency of occurrence of aftershocks decays rapidly according to the Omori's law:

$$n(t) = \frac{k}{(c+t)^{-p}}. \tag{1}$$

where $n(t)$ is the frequency of aftershocks at time t after the mainshock; k, c and p are constants that depend on the size of earthquake. The distribution in space of aftershocks is often related to the fault area or its length. There are a lot of empirical relations to estimate the fault area or the length fault [4] developing the empirical formula related to the fault area:

$$log(A) = 1.02M_s + 6.0. \tag{2}$$

Regarding to the fractured fault segment length L related to the seismic event magnitude Utsu [5] found the empirical relation:

$$log(L) = 0.5M_s - 1.8. \tag{3}$$

An important formula is the Gutenberg-Richter's relation [6] which connect the size and the frequency occurrence. They first proposed that the frequency of aftershocks occurrence can be represented by the formula:

$$log(N) = a - bM. \tag{4}$$

in a given period of time and in a given region. In the previous formula N is the number of earthquakes with a fixed magnitude M, a and b are constants characteristic for the given area. Concerning the occurrence of large aftershocks, in the first ten days, after a mainshock with magnitude greater than 7.0, 153 sequences all over the world from 1973 to 2004 have been analyzed , from the NEIC-USGS database (http://neic.usgs.gov/neis/epic/). It has been noticed that in the first ten days there is a probability about 81% to have a large aftershock (with magnitude $M \geq 5.5$) how is shown in Figure 1.

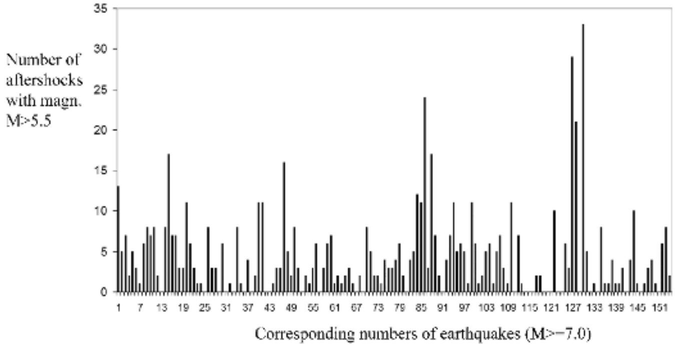


Fig. 1. The figure shows the number of aftershocks with magnitude $M > 5.5$ in the first 10 days (y axis) related to the 153 earthquakes having the magnitude of mainshock greater then 7.0 all over the word (x axis). The period is ranged from 1973 to 2004.

In the following, general criteria used by the authors to define a seismic sequence are reported. Firstly, it has been calculated the dimensions of the involved area using the Utsu [5] empirical relation (3) and so we acquired data in a square with sides at a distance $3L$ from the mainshock epicenter, where L represents the fault length of the mainshock. The data set is related to the events with magnitude $M \geq 1$ recorded in the computed square in one year after the mainshock. The evaluation of the completeness threshold, Mc, of a dataset is made using the Gutenberg-Richter relation by using the approach in detail described in [1]. It is computed with data related to the events with magnitude $M \geq 1$ reported by web site of NEIC-USGS databank (http://neic.usgs.gov/neis/epic/), in the first 10 days after the mainshock and in a square centered on the mainshock. Using the data concerning the first ten days of the sequence, the "barycenter" [7] of the aftershocks sequence has been calculated by means of:

$$B_{lat} = \frac{\sum_{i=1}^{n} Lat_i}{n} \qquad B_{lon} = \frac{\sum_{i=1}^{n} Lon_i}{n}. \tag{5}$$

where Lat_i and Lon_i are, respectively, the latitude and the longitude of the i-th aftershock epicenter, and n is the number of aftershocks with a magnitude $M \geq Mc$. The final rectangular sector is centered on the sequence "barycenter" and has its sides at a distance from this point, in terms of latitude and longitude, equals to $1.5L$. For the temporal duration d of the seismic sequence, $d = n1 + n2$, where $n1$ is the number of days equals to the number of aftershocks in the 24 hours starting from the occurrence of the mainshock, and $n2$ is the number of days, after $n1$, that reach and include 10 days in which there are no earthquake [1], [2].

3 About RBF Neural Networks

The Neural Networks derive from the idea to give the computer a sort of intelligence and ability to take, in general, some decisions. Principally they are systems that "learn" something to carry out correctly, for example, complicated relations, non-linear and multi-variable relations. A Neural Network is formed of some number of neurons and connections that simulate the behavior of biological activity and in this work we used the RBF Neural Networks. RBF Neural Networks (RBF-NNs) have capabilities to solve function approximation problems. RBF-NNs consist of three layers of nodes: more than input and output layers, RBF-NNs have a hidden layer, where Radial Basis Functions are applied on the input data [8]. A schematic representation of RBF-NN is described by Fig. 2, where R represents the number of elements in input vector, S1 the number of neurons in layer 1 and S2 represents the number of neurons in layer 2.

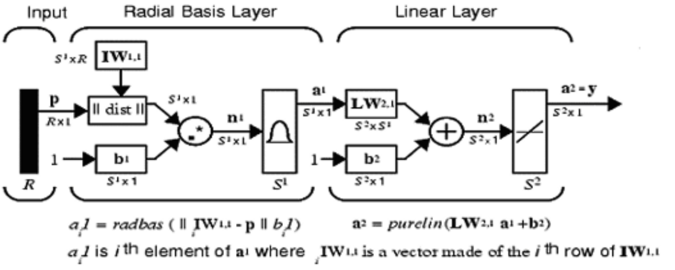

Input
Radial Basis Layer
Linear Layer
S¹xR
IW1,1
p
Rx1
|| dist ||
S¹x1
n¹
S¹x1
a¹
S¹x1
LW2,1
S²xS¹
n²
S²x1
a² = y
S²x1
1
b¹
S¹x1
1
b²
S²x1
R
S¹
S²
a¹ᵢ = radbas (|| ᵢIW1,1 - p || b¹ᵢ)
a² = purelin(LW2,1 a¹ +b²)
a¹ᵢ is i th element of a¹ where ᵢIW1,1 is a vector made of the i th row of IW1,1

Fig. 2. Schematic representation of a Radial Basis Function Neural Network

The $||dist||$ box in this figure accepts the input vector **p** and the input weight matrix $IW^{1,1}$, and produces a vector having S^1 elements. The elements are the distances between the input vector and vectors ${}_iIW^{1,1}$ formed from the rows of the input weight matrix. We can understand how this network behaves by following an input vector **p** through the network to the output a^2. If we present an input vector to such a network, each neuron in the radial basis layer will output a value according to how close the input vector is to each neuron's weight vector. Thus, radial basis neurons with weight vectors quite different from the input vector **p** have outputs near zero. These small outputs have only a negligible effect on the linear output neurons. In contrast, a radial basis neuron with a weight vector close to the input vector **p** produces a value near 1. If a neuron has an output of 1 its output weights in the second layer pass their values to the linear neurons in the second layer. In fact, if only one radial basis neuron had an output of 1, and all others had outputs of 0's (or very close to 0), the output of the linear layer would be the active neuron's output weights. This would, however, be an extreme case. Typically several neurons are always firing, to varying degrees. Now let us look in detail at how the first layer operates. Each neuron's weighted input is the distance between the input vector and its weight vector. Each neuron's net input is the element-by-element product of its weighted input with its bias. Each neuron's output is its net input passed through radial basis function. If a neuron's weight vector is equal to the input vector (transposed), its weighted input is 0, its net input is 0, and its output is 1. This work born from the idea to predict large aftershocks subsequently a strong earthquake with magnitude $M \geq 7.0$ using the RBF-NN and basing on the *Delta/Sigma* method [1], [2]. This method highlight some methodological aspect related to the observation of possible seismic anomalies in the temporal decay of aftershocks sequences that could be considered as precursors of a large aftershock. So RBF-NNs help us to predict the seismic temporal series after a training phase, in which an RBF-NN learns the trend of analyzed function by means of empirical data [9]. Therefore, it needs to build a training database (DBTrain), collecting input and output data, in order to carry out the training phase. The RBF-NN capabilities are then evaluated by a testing phase, in which a testing database (DBTest) is used to compare RBF-NN simulations with actual data. In next section, the implementation of both DBTrain and DBTest is described.

4 The Training and Testing Databases

In-Out relationship, i.e. the transfer function of the net, is obtained by a training process with empirical data. In practice the neural network learns the connection function through output and input by real examples of pairs In/Out. In fact, for every input presented to the net during the training process, the same provides an output that moves away of a certain quantity δ from the desired output. During the training phase, some parameters are modified to converge to the optimal solution. The mentioned parameters are the "weights" or the "connection factors" between the neurons composing the net. As described in the previous section, RBF-NN needs DBTrain in order to learn how to approximate the trend of $n(t)$. In this paper we considered the seismic sequences happened in California, Taiwan region and Colfiorito (Italy). Using the above mentioned criteria, it has been obtained for California sequence a completeness magnitude equal to 4.0, the magnitude of mainshock is equal 7.4 (lat 34.59N, -116.27E); for Taiwan sequence a completeness magnitude equal to 4.2, the magnitude of mainshock is equal 7.7 (lat 23.77N, 120.98E); for Colfiorito seismic sequence a completeness magnitude equal to 4.0, the magnitude of mainshock is equal 6.4 (lat 43.08N, 12.81E); the duration of the seismic sequences are respectively 40, 80, 39 days. Therefore, DBTrain has been implemented by $n(t)$ series concerning the earthquake occurred in California (USA) at October 16th 1999 (magnitude 7.4) and by $n(t)$ series concerning the earthquake occurred in Taiwan region [10] on September 20th 1999 (M = 7.7) (Fig. 3).

Using the California and Taiwan seismic sequences, trainDB has been implemented according to the following algorithm. Given: R inputs, Q outputs, D days

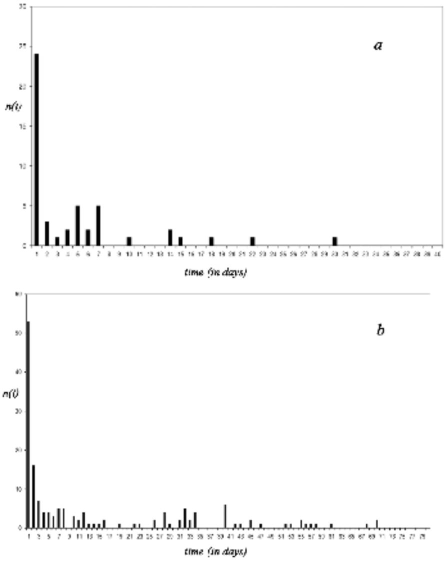


Fig. 3. a) Temporal trend of $n(t)$ for the California seismic sequence occurred on October 16th 1999. b) Temporal trend of $n(t)$ for the Taiwan seismic sequence occurred on September 20th 1999. It is possible to note that the two sequences are stopped after ten days without seismicity.

of useful ”time prediction window”, the discrete time serie n(k) which represents the considered statistics of daily aftershock (k=1, , K), then the set I = n(i), n(i+1), , n(i+R-1) represents the RBF-NN input database and O = n(i+R+D-1), n(i+R+D), , n(i+R+D+Q-2) is the RBF-NN outputs (i=1, , K-(R+D+Q-2)). With the specific application described in this paper, it has been considered 3 inputs, 1 output and a time prediction window equal to 3 days. Therefore, the triplet n(i), n(i+1), n(i+2) has been used in order to foresee n(i+5), with i=1, , K-6 and K is the summation of California and Taiwan sequence lengths (K=120). In a similar way, testDB has been implemented using only $n(t)$ of aftershock temporal series occurred in Colfiorito (Italy) on September 26th, 1997, having a mainshock's magnitude equal to 6.4 (K=39).

5 Results and Conclusions

In this work a Radial Basis Function Neural Network has been implemented in order to predict the trend of $n(t)$ for aftershocks temporal series, that is the numerical series of daily-earthquake's number occurred after a great earthquake with magnitude $M > 7.0$ Richter. According to the statistical method described in [1], aftershocks are foreseen 5 days after the possible occurrence. By adding a RBFNN model, it is possible to evaluate the $n(t)$ trend in an heuristic way, so estimating the behavior of the seismic sequence with a three day prediction.

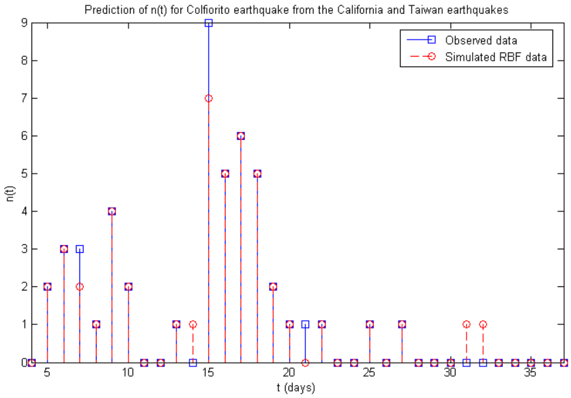


Fig. 4. Comparison between observed data trend (squares) and the simulated RBF-NN data trend (circles) for the Colfiorito seismic sequence

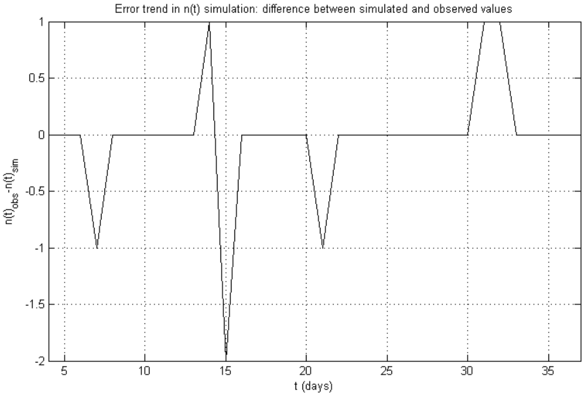


Fig. 5. Error trend from the comparison between observed data and RBF-NN's simulated data trends

The conjunction of proposed approach with the *Delta/Sigma* method allows to extended the prediction window up to 8 days before a possible aftershock occurrence. In particular, a RBF-NN has been implemented for the Colfiorito seismic sequence. Results of our experimentation are fairly satisfactory. The Root Mean Square Error amounts in percentage at 3.47%. Fig. 4 shows the results retrieved by RBF-NN simulation for the Colfiorito seismic sequence.

It is possible to obtain the errors trend from the comparison between observed data trend and the simulated RBF data trend as shown in Fig. 5. It is possible to denote how the error frequency decreases with days, since an aftershock event is day-by-day less probable.

Let us denote that depicted RBF-NN simulations starts from the fourth day of seismic sequence, that is the first output of RBF-NN. From the previously proposed figures, it is possible to note that the mean absolute error's value between real and simulated data of Colfiorito seismic sequence is equal to 1. It appears only one absolute error's value equal to 2 on the 15th day, but it could reenter in the norm because Fig. 4 shows that on the considered day there is an increment of the seismicity with respect to the theoretical trend. Therefore, it is possible to consider a good performance of RBF-NN prediction for $n(t)$.

References

1. Caccamo, D., Barbieri, F. M., D'Amico, S., Lagan, C., Parrillo, F.: The temporal series of the New Guinea 29 April 1996 aftershock sequence. P.E.P.I., Vol. 153, No. **4**, (2005) 175–180
2. Caccamo D., Barbieri F. M., Lagan C., D'Amico S., Parrillo F.: A study about the aftershock sequence of 27 December 2003 in Loyalty Islands. B.G.T.A., (2006) In press
3. Console R., Di Giovambattista R.: Local earthquake relative location by digital records, Phys. Of the Earth and Plan. Int., **47**, 1987 43–49
4. Utsu T. and Seki A.: A relation between the area of aftershock region and the energy of mainshock. J. Seismol. Soc. Jpn, **7**, (1954) 233–240
5. Utsu T.: Aftershocks and earthquake statistics (I) source parameters which characterize an aftershock sequence and their interrelations, J. Fac. Sci. Hokkaido Univ., Ser. VII, **3**, (1969) 129–195
6. Gutenberg B., Richter C. F.: Earthquake magnitude, intensity, energy and acceleration, Bull. Seism. Soc. Am., **32**, (1942) 162–191
7. D'Amico S., Caccamo D., Barbieri F. M., Lagan C.: Some methodological aspect related to the prediction of large aftershocks, E.S.C. 2004, XXIX General Assembly, book of abstracts, (2004) 132–133
8. Chen S., Cowan C. F. N. and Grant P. M.: Ortogonal Least Squares Learning Algorithm for Radial Basis Function Network. IEEE Transaction on Neural Networks, vol. 2, No. 2, (1991) 302–309
9. Christodoulou C. and Goergiopoulos M.: Application in Neural Networks in Electromagnetics. Artech House Publishers, Norwood Massachussetts (2001)
10. Parrillo F., D'Amico S., Cacciola M., Morabito F. C., Barrile V., Versaci M., Caccamo D.: Reti Neurali Radial Basis Function e metodo Delta/Sigma per la previsione di forti repliche. Atti del XXIV Convegno Gruppo Nazionale Geofisica della Terra Solida (2005) 197–200

Motion Vector Prediction Using Frequency Sensitive Competitive Learning

HyungJun Kim

Division of Information Technology
Hansei University, Korea
harry@hansei.ac.kr

Abstract. We propose a search region prediction method using a Frequency Sensitive Competitive Learning(FSCL) algorithm for the adaptive vector quantization of the motion vector. We train the motion vector codebook using the first two successive images of a sequence of images and utilize it for search region prediction. The proposed method can reduce computation time by using a smaller number of search points compared to other methods, and also decreases the bits required to represent motion vectors. The experimental results show that it provides competitive PSNR values compared to other block matching algorithms.

1 Introduction

We propose a new method for estimating motion vectors in an image sequence. The proposed method predicts the search region by using Frequency Sensitive Competitive Learning Vector Quantization(FSCL-VQ) and evaluates distortion for the predicted points. The motion estimation and compensation techniques have been widely used in video compression due to its capability of reducing the temporal redundancies between frames. One of the algorithms developed for motion estimation is block-based technique, also known as Block-Matching Algorithm(BMA). In this technique, the current frame is divided into fixed size of blocks, then each block is compared with candidate blocks in reference frame within search area. System performance depends on how accurate the motion vectors are estimated. The full search method that matches all points in the search area must be used to detect the motion vectors more accurately; however, it requires much computation and hardware complexity.

The accuracy of the prediction can usually be improved by compensating for motion between the reference frame and the current frame[1]. Since the temporal correlation as well as the spatial correlation is very high in moving pictures, a high compression ratio can be achieved by using the Motion Compensated Coding(MCC) technology. MCC consists of a motion compensating by the precise motion estimation and prediction error encoding part[2]. In motion compensated prediction coding method with BMA, the amount of information for motion vectors and prediction error must be as small as possible. The size of search area may be adjusted depending on the displaced block results and the block classification information in the successive frames of the block[3].

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 917–924, 2006.

2 Search Region Prediction Using Vector Quantization

The simplest method of temporal prediction is to use the previous frame as the predictor for the current frame. Block matching algorithms are utilized to estimate motion at a block of pixels. This block of pixels is compared with a corresponding block within a search region in the previous frame. The process of BMA divides an image into fixed size sub-images, and then finds one dominant match for the previous frame by maximizing cross correlation. In a typical BMA, the current frame of a video sequence is divided into non-overlapping square blocks of pixels such as of size $N \times N$. For each reference block in the current frame, BMA searches for the best matched block within a search window of size $(2W + N) \times (2W + N)$ in the previous frame, where W represents the maximum allowed displacement. Then the relative position between the reference and its best matched block is acquired as the motion vector of the reference block. A non-negative matching error function $D_p(i,j)$ with metric dimension p is defined over all the positions to be searched, i.e.,

$$D_p(i,j) = \sum_{m=1}^{N} \sum_{n=1}^{N} |I_t(l+m, k+n) - I_{t-1}(l+m+i, k+n+j)|^p, \\ -W \leq i,j \leq W, \quad p = 1 \text{ or } 2 \tag{1}$$

where $I_t(l,k)$ is the reference block of its upper left pixel at the coordinate (l,k) in the current frame, and $I_{t-1}(l+i, k+j)$ is a candidate block of its upper left pixel at the coordinate $(l+i, k+j)$ in the previous frame. The value (i,j) is displacement which minimizes the $D_p(i,j)$. Even though motion vector detection schemes using BMA have been widely utilized, they have many drawbacks. For instance, they assume that all the pixels within the block have uniform motion because they detect motion vectors on a block-by-block basis. This assumption is acceptable for small block sizes (8×8 or 16×16). However, having a smaller block-size increases the number of blocks and requires higher transmission rate because of an increase in the amount of motion vectors to be transmitted[4,5]. Therefore, there is a tradeoff in image quality associated with faster motion compensation schemes; longer processing time is required to find the vector field for providing higher image quality.

For Vector Quantization(VQ), the space of vectors to be quantized is divided into a number of regions. A reproduction vector is calculated for each region. Given any data vector to be quantized, the region in which it lies is determined and the vector is represented by the reproduction vector for that region. More formally, vector quantization is defined as the mapping of arbitrary data vectors to an index m. Thus, the VQ is mapping of k-dimensional vector space $\mathbf{x} = (x_1, x_1, \ldots, x_k)$ to a finite set of symbols $m \in \{M\}$. Assuming a noiseless transmission or storage channel, m is decoded as $\mathbf{x}$. The collection of all possible production vectors is called the codebook. In general, this requires knowing the probability distribution of the input data. Typically, however, this distribution is unknown, and the codebook is constructed through process called training.

During the training, a set of data vectors that is representative of the data that will be encountered in practice is used to determine an optimal codebook[6].

The training and encoding processes are computationally expensive. Moreover, most of the algorithms currently used for VQ design are batch mode algorithms, and need to have access to the entire training data set during the training process. Also, in many communication applications, changes in the communication channel mean that a codebook designed under one condition is inappropriate for use in another condition. Under these circumstances, it is much more appropriate to work with adaptive VQ design methods, even if they are suboptimal in a theoretical sense. Another benefit of formulating VQ using a neural network is that a number of neural network training algorithms such as Competitive Learning(CL) and FSCL can be applied to VQ. FSCL is particularly effective for adaptive VQ in image compression systems[7].

3 Motion Vector Estimation by Search Region Prediction

The performance of motion vector detection can be increased because motion vectors conventionally have high a spatiotemporal correlation. We propose a new motion vector estimation technique utilizing this correlation. Assume that the neural network VQ is to be trained on a large set of training data. Furthermore assume that the weight vectors $\mathbf{W}_i(n)$ are initialized with random values. The algorithm for updating the weight vectors is as follows. The input vector is presented to all of the neural units and each unit computes the distortion between its weight and the input vector. The unit with the smallest distortion is designated as the winner and its weight vector is adjusted towards the input vector. Let $\mathbf{W}_i(n)$ be the weight vector of i^{th} neural unit at the i^{th} iteration, then the basic CL algorithm can be summarized as follows:

$$z_i = \begin{cases} 1 & \text{if } d(\mathbf{x}, \mathbf{W}_i(n)) = \min_{1 \le j \le M} d(\mathbf{x}, \mathbf{W}_j) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $d(\mathbf{x}, \mathbf{W}_i(n))$ is the distance in the L_2 metric between the input vector $\mathbf{x}$ and the coupling weight vector $\mathbf{W}_i(n)$, and z_i is its output. Under-utilization problem may occur in CL which means some of the neurones are left out of the learning process and never win the competition. The new weight vectors $\mathbf{W}_i(n+1)$ are computed as:

$$\mathbf{W}_i(n+1) = \mathbf{W}_i(n) + \alpha(\mathbf{x} - \mathbf{W}_i(n))z_i \tag{3}$$

where the parameter α is the learning rate, and is typically reduced monotonically to zero as the learning progresses. A problem with this kind training procedure is that it occasionally leads to under-utilized neural units. FSCL algorithm has been suggested to overcome a drawback of CL. FSCL algorithm addresses the problem by keeping a record of how frequent each neurone is the winner to maintain that all neurones in the network are updated an approximately equal number of times. In the FSCL network, each unit incorporates a count of the

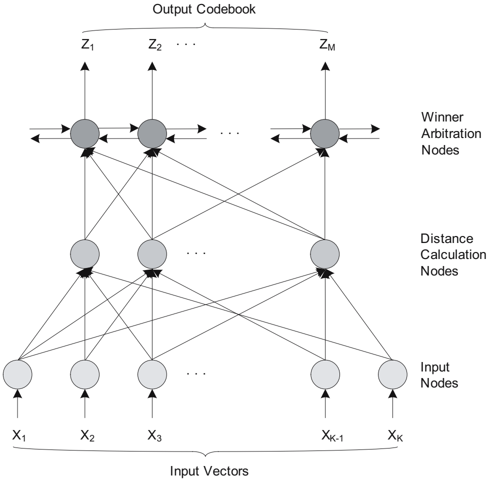


Fig. 1. The architecture of FSCL algorithm

number of times it has been the winner. A modified distortion measure for the training process is defined as follows:

$$d^*(\mathbf{x}, \mathbf{W(n)}_i) = d(\mathbf{x}, \mathbf{W}_i(n))u_i(n) \qquad (4)$$

where $u_i(n)$ is the total number of times for neurone i up to the i^{th} training cycle. Hence, the more the i^{th} neurone wins the competition, the greater its distance from the next input vector. Therefore, the chance of winning the competition diminishes. This way of tackling the under-utilization problem does not provide interactive solutions in optimizing the codebook. The winning neural unit at each step of the training process is the unit with the minimum d^*. Fig. 1 illustrates the architecture of FSCL algorithm.

Fig. 2 describes the block diagram of the proposed motion vector estimation method using a neural networks vector quantizer with BMA. We find the motion vectors using the full search block matching algorithm from two successive frame images and train a codebook. The codebook is used as the predicted search region. First, we find motion vectors using the full search method from the training images and then, train the neural network vector quantizer codebook using these motion vectors. Second, a motion vector can be estimated using the codebook as a motion prediction region. Codebook retraining procedure may be needed if the distortion measure is higher than a certain threshold value. Fig. 3 represents examples of the initial codebook and the output codebook that has 25 codewords. The codewords in the codebook represent the motion vectors for the input image sequences. Since the codebook is used as the search region for estimating the motion vectors, the search points and computation can be reduced compared with the full search BMA. In addition, the information required to transmit the motion vectors can be reduced. The computational cost is also improved because the number of search point is reduced. Based on the

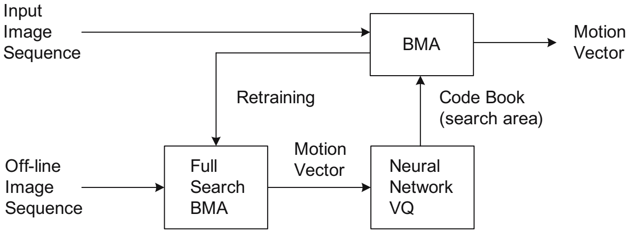


Fig. 2. Block diagram of the proposed motion vector estimation system

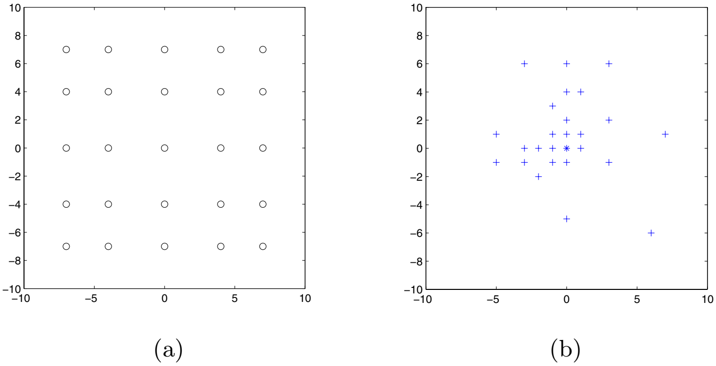


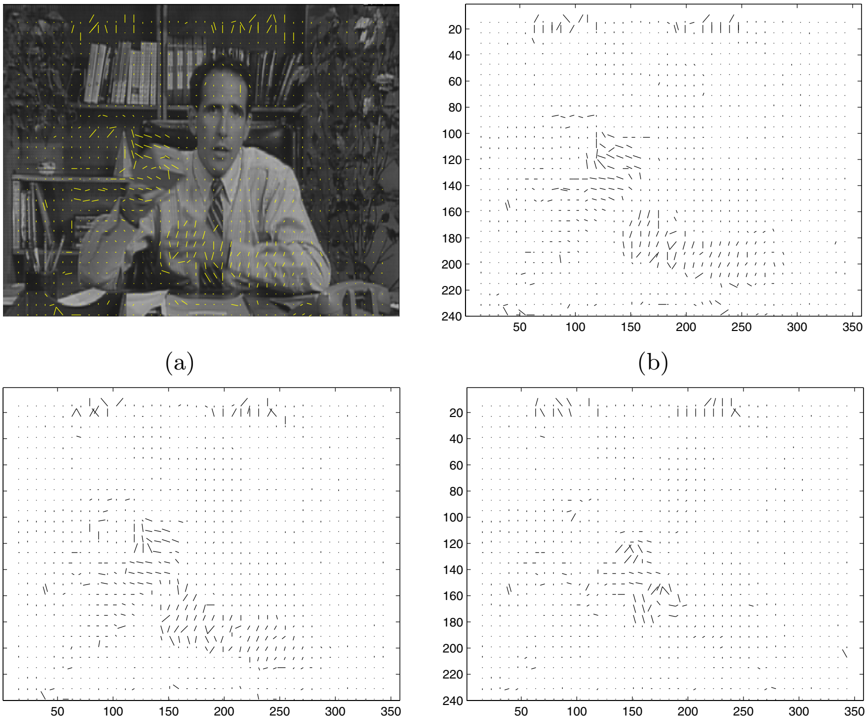
Fig. 3. Examples of (a) initial value of codebook and (b) trained value of codebook

above motion vectors as the training input data, the codebook is designed with the FSCL algorithm.

4 Experimental Results

The SIF version of Salesman and Flower garden image sequences were used for the experiment. The size of an SIF sequence is half of its CCIR 601 version in both dimensions. The block size for BMA was set to 8×8. Since the recommended search region by MPEG is 15 pixels in both horizontal and vertical directions, we choose a search region of ±7 pixels in both spatial directions. We also set a codebook size of 64 motion vectors for this particular experiments.

In this research, we used Peak Signal to Noise Ratio(PSNR) as objective quality measure. Fig. 4 shows examples of motion vector map for the 16$^{\text{th}}$ frame of Salesman image sequence using full search, TSS, and the proposed methods. We set the block size to be 4, the search area to be ±7 pixels, and exaggerating motion vectors to make the effect more visible. It demonstrates that the smoothing effect of the proposed methods is superior to other methods. Using the smoothing effect, we can eliminate errors which may be caused by quantization process of motion vectors and can also reduce the number of bits required to represent motion vectors. Fig. 5 represents the PSNR values of the first and the last 30



(a)

(b)

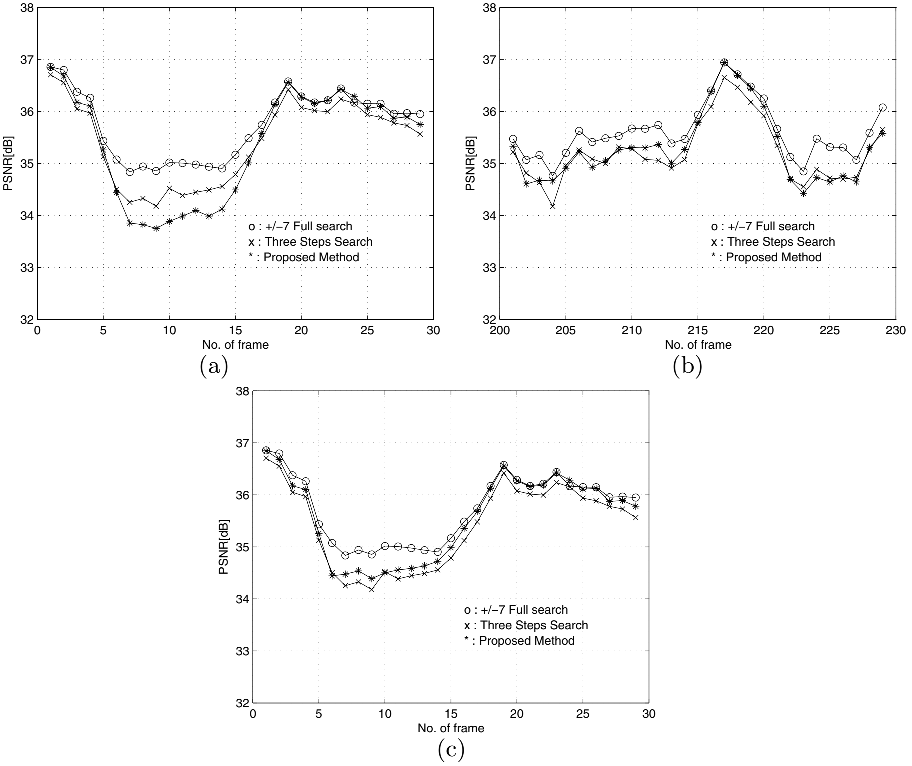
(c)

(d)

Fig. 4. Examples of motion vector map(8 × 8 blocks) for the 16th frame of Salesman image sequence: (a) the 16th frame with motion vector map, (b) full search, (c) TSS, and (d) the proposed method

frames of which the smoothing effect of motion vectors have been calculated using three different methods. We can also find an improvement with codebook retraining compared to no retraining at the frame 7 where a big movement has occurred as shown in Fig. 5.

Table 1 presents the number of search points and the average PSNR of the first ($1^{\text{st}} \sim 30^{\text{th}}$) and last ($201^{\text{st}} \sim 230^{\text{th}}$) 30 frames. We compare the performance of the proposed method with that of BMA with full search(the search region is 15 pixels in both horizontal and vertical directions) and Three Step Search(TSS). Note that the PSNR(1) value for the first 30 frames are after codebook retraining. As shown in Table 1, the number of possible motion vectors for BMA with full search is 225, which requires about 8 bits per a motion vector for fixed length encoding. Therefore, we have compressed the number of motion vectors from 225 to 25 or from 8 bits to 5 bits per vector. The number of search points requires for the proposed method is smaller than that of full search method while having almost the same average PSNR values. The number of matches for the proposed method is a bit lower compared to TSS method and the average PSNR values are mostly higher than that of TSS method.



(a)

(b)

(c)

Fig. 5. Performance comparison for Salesman image sequence: (a) frames from 1 to 30, (b) frames from 201 to 230, and (c) codebook retraining at the frame 7

Table 1. Performance comparison for two sets of 30 frames of Salesman(1) and Flower garden(2) image sequence

Method	Frames	Search Pt.	Bits/MV	PSNR(1)(dB)	PSNR(2)(dB)
Full search(±7)	1 ∼ 30	225	8	35.75	34.40
	201 ∼230	225	8	35.61	34.81
TSS	1 ∼ 30	27	5	35.42	33.52
	201 ∼230	27	5	35.22	33.86
Proposed	1 ∼ 30	25	5	35.47	34.21
	201 ∼230	25	5	35.30	34.13

5 Conclusions

We proposed a new search region prediction method for motion estimation. We found motion vectors using the full search Block Matching Algorithm(BMA) from the initial image sequences, and trained Frequency Sensitive Competitive Learning(FSCL) to design a codebook. We then utilized that codebook for the motion estimation. The proposed method uses the spatial correlation of motion vectors in image sequences, therefore reducing search area, decreasing bits required to transmit motion vectors, and increasing the compression rate. The proposed method achieves almost the same PSNR value as full search method, and also requires the least number of search points and bits for motion vectors. The computer simulations show that the proposed method is competitive to the full search and the Three Step Search(TSS) methods. Codebook retraining procedure may be needed if there is a big movement and therefore the distortion measure is higher than a certain threshold value. In real communication applications, a codebook designed under one condition is inappropriate for use in another condition. Thus, it is appropriate to work with adaptive Vector Quantization(VQ) methods, even if they are suboptimal in a theoretical sense. When we encounter an unexpected sudden movement, an additional effort is necessitated to retrain the codebook. The foremost reason for big movement during this experiment was because we only used the first two frames for the initial training. Therefore, enlarging training set will possibly eradicate the retraining process. Enhancing the initial training performance is our most current project.

Acknowledgements

This work was supported by Hansei University.

References

1. Li, B., Li, W., and Tu, Y.: A fast block-matching algorithm using smooth motion vector field adaptive search technique. Journal of Computer Science and Technology, 18(1), (2003) 14–21
2. Yao, W., Wenger, S., Jiantao, W., and Katsaggelos, A. K.: Error resilient video coding techniques. IEEE Signal Processing Magazine, 17(4), (2000) 61–82
3. Oh, H. S. and Lee, H. K.: Block-matching algorithm based on an adaptive reduction of the search area for motion estimation. Real-Time Imaging, 6, (2000) 407–414
4. Iinuma, K., Koga, T., Niwa, K. and Iijima, Y.: A motion-compensated interframe codec. in Proc. Image Coding, SPIE, 594, (1985) 194–201
5. Lee, Y. Y. and Woods, J. W.: Motion vector quantization for video coding. IEEE Trans. Image Processing, 4(3), **Mar.**, (1995)
6. Gersho, A. and Gray, R. M.: Vector Quantization and Signal Compression. Kluwer Academic Publishers, (1991)
7. Al Sayeed, C. and Ishteak Hossain, A.: Image compression using frequency-sensitive competitive neural network. in Proceedings of the SPIE, 5637, (2005) 611–618

Forecasting the Flow of Data Packets for Website Traffic Analysis – ASVR-Tuned ANFIS/NGARCH Approach

Bao Rong Chang[1,*], Shi-Huang Chen[2], and Hsiu Fen Tsai[3]

[1] Department of Computer Science and Information Engineering
National Taitung University, Taiwan 950
Phone: +886-89-318855 ext. 2607; Fax: +886-89-350214
[2] Department of Computer Science and Information Engineering
[3] Department of International Business
[2,3] Shu-Te University, Kaohsiung, Taiwan 824
brchang@nttu.edu.tw, shchen@mail.stu.edu.tw,
soenfen@mail.stu.edu.tw

Abstract. Forecast of the flow of data packets between client and server for a website traffic analysis is viewed as a part of web analytics. Thousands of web-smart businesses depend on web analytics to improve website conversions, reduce marketing costs, website optimization, website monitoring and provide a higher level of service to their customers and partners. This paper particularly intends to develop a high-accuracy prediction approach as the need for a website traffic analysis. The proposed composite model (ASVR-ANFIS/NGARCH) is schemed to build a systematic structure such that it is not only to improve the predictive accuracy because of resolving the problems of the overshoot and volatility clustering simultaneously, but also to boost website tracking capacity helping each webmaster to optimize their website, maximize online marketing conversions and lead campaign tracking.

1 Introduction

Webmaster in fact does not want to spend money or time on a website that sits idle not yielding an enquiry or a sale. In other words, webmaster want apply website analytics (or free counter) to design websites that are integrated with effective search engine marketing strategies to generate traffic and convert that traffic to sales [1]. What we need is to seek the website analytics that provides detailed return-on-investment analysis for an unlimited number of search engine advertising, banner advertising, affiliate marketing or email marketing campaigns and click in- and out tracking, combined with website statistics [2]. Therefore, website traffic analysis has become the trusted standard task in website statistics for various internet companies such as travel, dating sites and online shops. This is because website tracking capacity

[*] Corresponding author.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 925 – 933, 2006.

will help each webmaster to optimize their website, maximize online marketing conversions and lead campaign tracking. However, the website tracking or traffic analysis is related to the flow of data packets between hosts. In particularly, a look-ahead prediction of the flow of data packets is considered as a measure to guess the possibility of big or small fluctuation over data flow instantly, in such a way that webmaster or website analytics software can in-time adjust the current website resources dynamically because of a prior-sign of changes in data-packet flow toward webmaster or website analytics software. That is, the forecast of the flow of data packets among hosts would make a great help to analyze the website traffic and in the mean time facilitate the website tracking operation. The forecast of inflow and outflow of data packets among hosts can be realized by employing a non-periodic short-term predictor and such type of predictor samples the most recent data to generate informative signal, namely, a prior-sign of changes in data-packet flow.

Several well-known forecast models have challenged a few crucial problems. For example, grey model (GM) [3] has encountered the overshoot problem such that it will induce big residual errors around turning-point region in time series during the forecast. Moreover, autoregressive moving-average (ARMA) [4], artificial neural network (ANN) [5], or adaptive neuro-fuzzy inference system (ANFIS) [6] model can not avoid the volatility clustering [7] and thus this effect deteriorates the predictive accuracy a lot for the non-periodic short-term forecast. Therefore, in this paper, incorporating a nonlinear generalized autoregressive conditional heteroscedasticity (NGARCH) [8] model into ANFIS system is schemed to tackle the overshoot and volatility clustering effects at the same time during single-step-look-ahead prediction. The proposed composite model ANFIS/NAGRCH is tuned optimally by adaptive support vector regression (ASVR) [9] to form a linear combination of both models in such a way that it is not only simplify the complex system practically, but also improve the predictive accuracy significantly because of resolving the problems of the overshoot and volatility clustering simultaneously. In short, in order to manage the web resources effectiveness and efficiency, a higher accurate prediction is required to forecast the in-flow and out-flow data packets applied for website traffic analysis. Web traffic analysis provides valuable information for web site administrators to customize the information that is hosted on their web servers so as to reach a larger audience.

2 ARMAX/NGARCH Composite Model

ARMAX/NGARCH composite model allows you to deal with the presence of conditional heteroscedasticity for time series prediction, especially in financial time series applications like asset return problem. The ARMAX [10] encompass autoregressive (AR), moving-average (MA), and regression (X) models, in any combinations as expressed below.

$$y_{armax}(t) = C^{armax} + \sum_{i=1}^{r} R_i^{armax}\, y(t-i) + e_{resid}(t) + \\ \sum_{j=1}^{m} M_j^{armax}\, e_{resid}(t-j) + \sum_{k=1}^{N_x} \beta_k^{armax}\, \mathbf{X}(t,k) \quad , \tag{1}$$

The NGARCH(p,q) [11] consists of nonlinear time-varying conditional variances and Gaussian innovations. Its mathematical formula is shown as follows.

$$\sigma_{ntvcv}^2(t) = K^{ng} + \sum_{i=1}^{p} G_i^{ng}\sigma_{ntvcv}^2(t-i) + \sum_{j=1}^{q} A_j^{ng}\sigma_{ntvcv}^2(t-j)\left[\frac{e_{resid}(t-j)}{\sqrt{\sigma_{ntvcv}^2(t-j)}} - C_j^{ng}\right]^2 \tag{2}$$

with constraints

$$\sum_{i=1}^{p} G_i^{ng} + \sum_{j=1}^{q} A_j^{ng} < 1\,,\ K^{ng} > 0\,,\ G_i^{ng} \geq 0,\ i = 1,\ldots,p\,,\ A_j^{ng} \geq 0,\ j = -1,\ldots,q$$

3 Adaptive Support Vector Regression

We consider approximating functions $f(\cdot)$ solved by support vector regression (SVR) [12] with the form of

$$f(\mathbf{x},\mathbf{w}) = \sum_{i=1}^{l} w_i \phi(x_i) + b\,, \tag{3}$$

where $\phi(\cdot)$, w_i, and b denote a nonlinear mapping, a weighted value, and a bias, respectively. Furthermore, Vapnik introduced a general type of loss function, namely the linear loss function with ε-insensitivity zone [12], as

$$\left|y - f(\mathbf{x},\mathbf{w})\right|_{\varepsilon} = \begin{cases} 0 & if\left|y - f(\mathbf{x},\mathbf{w})\right| \leq \varepsilon, \\ \left|y - f(\mathbf{x},\mathbf{w})\right| - \varepsilon & otherwise \end{cases} \tag{4}$$

According to the learning theory of SVMs [13], this can be expressed by maximizing dual variables Lagrangian $L_d(\boldsymbol{\alpha},\boldsymbol{\alpha}^*)$ where l, $\mathbf{x_i}$, $\mathbf{y_i}$, and $K(\cdot,\cdot)$ denote the number of vectors, an input vector, an output vector, and the kernel function, respectively.

$$L_d(\boldsymbol{\alpha},\boldsymbol{\alpha}^*) = -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(\mathbf{x_i},\mathbf{x_j}) - \varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) - \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\mathbf{y_i}, \tag{5}$$

subject to the constraints

$$\sum_{i=1}^{l}\alpha_i = \sum_{i=1}^{l}\alpha_i^*\ ,\ 0 \leq \alpha_i \leq C,\quad i = 1,\ldots,l\,,\ 0 \leq \alpha_i^* \leq C,\quad i = 1,\ldots,l\ . \tag{6}$$

After obtaining the Lagrange multipliers α_i and α_i^*, we find the optimal weights of regression $\mathbf{w_0}$ and an optimal bias $\mathbf{b_0}$

$$\mathbf{w_0}=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)\phi(\mathbf{x_i})\,,\;\; \mathbf{b_0}=\frac{1}{l}\left(\sum_{i=1}^{l}(\mathbf{y_i}-\phi(\mathbf{x_i})^T\,\mathbf{w_0})\right). \tag{7}$$

A fast algorithm applied to constraint optimization for support vector regression (SVR) is called adaptive support vector regression (ASVR) [9]. It is designed for exploring three free parameters C, ε and σ_{rbkf} [9] such that the computation time of quadratic programming (QP) is significantly reduced and achieved rapid convergence to the near-optimal solution. An order n, see [9], is also predetermined and used in computing the free parameter C in Eq. (9) and Eq. (10). In this manner, a straightforward parameter-seeking is done rather than using a heuristic method with a long time for searching. Note that Eq. (9) and Eq. (10) are based on the modified Bessel function of second kind with the order n [14] as follows:

$$\varepsilon=v\times\frac{\left|\max(\mathbf{X})-\min(\mathbf{X})\right|}{2} \tag{8}$$

$$C=K_n(\varepsilon)=(-1)^{n+1}\{\ln(\varepsilon/2)+\gamma\}I_n(\varepsilon)+\frac{1}{2}\sum_{k=0}^{n-1}(-1)^k(n-k-1)!(\varepsilon/2)^{2k-n}+\frac{(-1)^n}{2}\sum_{k=0}^{\infty}\frac{(\varepsilon/2)^{n+2k}}{k!(n+k)!}\{\Phi(k)+\Phi(n+k)\} \tag{9}$$

$$I_n(\varepsilon)=\sum_{k=0}^{\infty}\frac{(\varepsilon/2)^{n+2k}}{k!\Gamma(n+k+1)} \tag{10}$$

where $\gamma=0.5772156...$ is Euler's constant and $\Phi(p)=1+\frac{1}{2}+\frac{1}{3}+...+\frac{1}{p}$, $\Phi(0)=0$ [14]. Eq. (11) determines a free parameter σ_{rbkf} of radial basis kernel function for quadratic programming in SVR.

$$\sigma_{rbkf}=\upsilon\cdot\sqrt{\sum_{i=1}^{l}(x_i-\bar{x})^2\Big/ l-1},\;\; \bar{x}=\sum_{j=1}^{l}x_j\Big/ l \tag{11}$$

4 ASVR Adaptation to Composite Model ANFIS/NGARCH

A single-step-look-ahead prediction can be implemented by adding a variation to the current observed datum [15], and the variation is defined as backward–difference as follows.

$$\hat{o}(k+1)=o(k)+\delta\hat{o}(k+1) \tag{12}$$

$$\begin{aligned}&\delta\hat{o}(k+1)\\&=f(o(k),o(k-1),...,o(k-s),\delta o(k),\delta o(k-1),...,\delta o(k-s))\end{aligned} \tag{13}$$

$$\delta o(k)=o(k)-o(k-1) \tag{14}$$

where $\hat{o}(k+1)$, $o(k)$, $\delta\hat{o}(k+1)$, and $\delta o(k)$ stand for the predicted output at next period, the current true observed datum, the predicted variation at next period, and the current variation, respectively.

we formulate a function of ANFIS output $\delta\hat{o}_{anfis}(k+1)$ and square-root of nonlinear conditional heteroscedasticity $\hat{\sigma}(k+1)$.

$$\widehat{\delta\omega}_{anfis/ngarch}(k+1)=f(\widehat{\delta\omega}_{anfis}(k+1),\,\hat{\sigma}_{ngarch}(k+1)) \tag{15}$$

For the simplicity, a linear weighted-average [16] is used to combine ANFIS output $\widehat{\delta\omega}_{anfis}(k+1)$ and NGARCH output $\hat{\sigma}_{ngarch}(k+1)$ as an approximation, $\widehat{\delta\omega}_{anfis/ngarch}(k+1)$. We denote this composite model as ANFIS/NGARCH.

$$\widehat{\delta\omega}_{anfis/ngarch}(k+1)=w_{anfis}\cdot\widehat{\delta\omega}_{anfis}(k+1)+w_{ngarch}\cdot\hat{\sigma}_{ngarch}(k+1) \tag{16}$$

ASVR is hence employed to tune both weights, w_{anfis} and w_{ngarch} , in ANFIS/NGARCH. This proposed approach is called ASVR-tuned ANFIS/NGARCH as shown in Fig. 1.

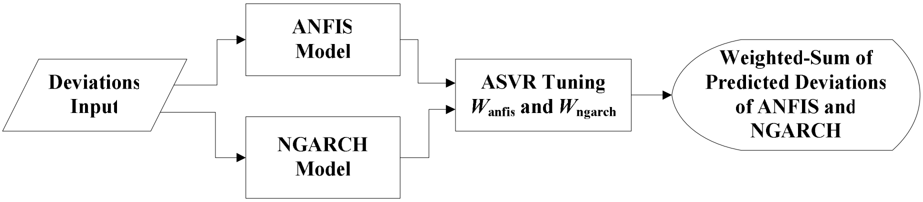

Deviations Input
ANFIS Model
NGARCH Model
ASVR Tuning W_{anfis} and W_{ngarch}
Weighted-Sum of Predicted Deviations of ANFIS and NGARCH

Fig. 1. Diagram of ASVR-tuned ANFIS/NGARCH outputs

5 Experimental Results and Discussions on Website Tracking

As shown in Fig. 2 to Fig. 5, six models, which are grey model (GM), auto-regression moving-average (ARMA), radial basis function neural network (RBFNN), generalized autoregressive conditional heteroscedasticity (GARCH), adaptive neuro-fuzzy inference system (ANFIS), and ANFIS with nonlinear conditional heteroscedasticity tuned by adaptive support vector regression (ASVR-ANFIS/NGARCH), are applied to forecasting (a) inflow data packets and (b) outflow data packets. Two experiments implemented at computer center of National Taitung University (NTTU) and Shu-Te University (STU), where NTTU is located in Taitung (eastern Taiwan) and STU in Kaohsiung (western Taiwan). Both inflow and outflow data packets are designed to measure the average bits per second in every 5 minutes for 280 sampling points at NTTU and STU [17][18], stared at 16:30 Aug. 5, 2004. First, we used the first 123 samples out of data set to train ANFIS, NGARCH, and ASVR concurrently so that the requisite parameters w_{anfis} and w_{ch} in Eq. (16) can be determined to build a prediction model. Next, the prediction is made for the rest sampling points (157 points) to forecast the inflow and outflow data packets. The inflow-data-packet and outflow-data-packet prediction at NTTU and STU implemented as shown in Figs. 2 and 3 as well as Figs. 4 and 5, respectively. The performance comparisons at NTTU and STU among six methods for the inflow-data-packet and outflow-data-packet predictions are listed in Tables 1 and 2 as well as Tables 3 and 4, respectively. The goodness of fit for the proposed approach on the inflow-data-packet and outflow-data-packet predictions at NTTU and STU are also tested successfully by Q-test [19] due to p-value (0.1572 and 0.2348 for NTTU) and (0.1692 and 0.3045 for STU) greater than level of significance (0.05). Next, we have checked the criteria [20] of (a) mean-square-error

(MSE) and (b) mean-absolute-percent-error (MAPE) as the performance evaluation among six methods for the experiments. As listed in Tables 1, 2,3, and 4, the proposed ASVR-tuned ANFIS/NGARCH approach achieves the best accuracy of prediction as well as obtain the satisfactory results due to no more big residual errors in prediction as shown in Figs. 2, 3, 4, and 5.

This study gives us the insight into the problem of forecasting the flow of data packets for website traffic analysis. First, the number of enrolled students at NTTU and STU are 3026 and 10325, respectively. According to the enrolled students, the use of internet at STU will be three times as many as that of internet at NTTU. However, we have checked the use of internet on both institutes and concluded a fact that the use of internet at STU is just about two times higher than NTTU. This is because one-third of STU students got the class at night, that is, actually they have enrolled for the extension education. Those students do not have much time to use internet at computer center or lab sufficiently because they stay with a short period of time in campus during class. In contrast, the graduate students are around 30% of total enrolled students at NTTU. Those people prefer staying in campus longer, and thus got much time to use internet at computer center or lab for their works. This fact implies that utilizing much more available internet bandwidth at STU can be exploited to extend website services such as money reports about pageviews, visitors, navigation, web site statistics, e-commerce tracking and online marketing campaigns that are supervised by webmaster or website analytics software. Instead, managing limited available internet bandwidth at NTTU is guided to help with optimizing your search engine marketing as well as SEO (Search Engine Optimizer) and PPC (Pay Per Click) marketing campaigns.

Table 1. The mean-square-error (MSE) and mean-absolute-percent-error (MAPE) between the desired values and the predicted results for inflow-data-packet at computer center of NTTU

Methods	Average MSE (unit: 10^3)	Average MAPE
GM	1.0022	0.0527
ARMA	14.672	0.6579
RBFNN	0.7451	0.0356
GARCH	1.2404	0.0485
ANFIS	1.0285	0.0458
ASVR-ANFIS/NGARCH	0.6875	0.0336

Table 2. The mean-square-error (MSE) and mean-absolute-percent-error (MAPE) between the desired values and the predicted results for outflow-data-packet at computer center of NTTU

Methods	Average MSE (unit: 10^3)	Average MAPE
GM	1.1172	0.0741
ARMA	2.8091	0.0985
RBFNN	1.1474	0.0402
GARCH	1.2599	0.0453
ANFIS	1.0705	0.0438
ASVR-ANFIS/NGARCH	0.6737	0.0302

Table 3. The mean-square-error (MSE) and mean-absolute-percent-error (MAPE) between the desired values and the predicted results for inflow-data-packet at computer center of STU

Methods	Average MSE (unit: 10^3)	Average MAPE
GM	0.8818	0.0454
ARMA	12.199	0.7618
RBFNN	0.9306	0.0386
GARCH	1.2471	0.0412
ANFIS	1.5456	0.0436
ASVR-ANFIS/NGARCH	0.6875	0.0327

Table 4. The mean-square-error (MSE) and mean-absolute-percent-error (MAPE) between the desired values and the predicted results for outflow-data-packet at computer center of STU

Methods	Average MSE (unit: 10^3)	Average MAPE
GM	0.9673	0.0625
ARMA	7.0605	0.3878
RBFNN	0.8231	0.0489
GARCH	1.2647	0.0562
ANFIS	1.7288	0.0436
ASVR-ANFIS/NGARCH	0.5339	0.0293

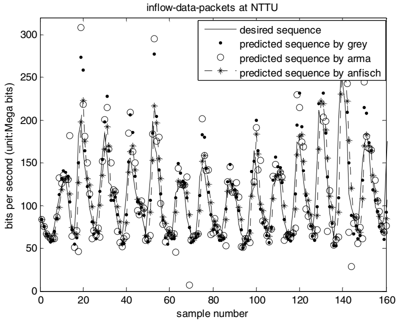


Fig. 2. Forecast of the inflow-data-packets at computer center of NTTU

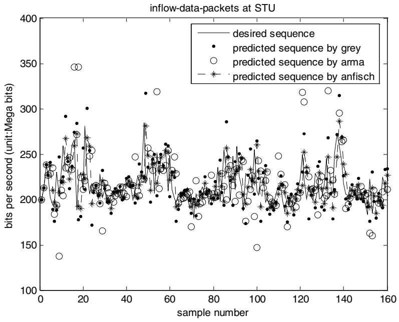


Fig. 4. Forecast of inflow-data-packet at computer center of STU

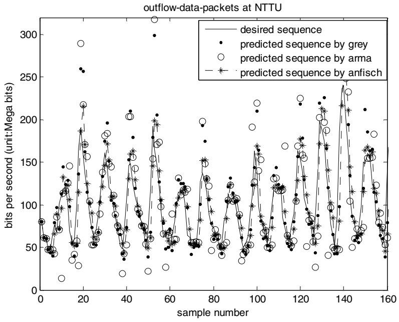


Fig. 3. Forecast of the outflow-data-packet at computer center of NTTU

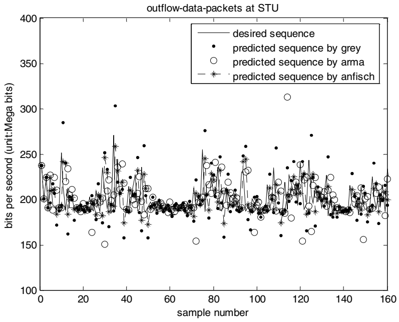


Fig. 5. Forecast of the outflow-data-packet at computer center of STU

6 Conclusions

Web traffic analysis provides valuable information for website administrators to customize the information that is hosted on their web servers so as to reach a larger audience. Thus, in order to manage the web resources effectiveness and efficiency, a higher accurate prediction is required to forecast the inflow and outflow data packets applied for website traffic analysis. In this paper, the proposed composite model (ASVR-ANFIS/NGARCH) is schemed to build a systematic structure such that it is not only to improve the predictive accuracy because of resolving the problems of the overshoot and volatility clustering simultaneously, but also to boost website tracking capacity helping each webmaster to optimize their website, maximize online marketing conversions and lead campaign tracking. We conclude that the proposed method can get the satisfactory results on the inflow and outflow prediction of data packets between server and clients.

Acknowledgements

This work is fully supported by the National Science Council, Taiwan, Republic of China, under grant number **NSC 93-2218-E-143-001**.

References

1. Funkhouser, T. A., Sequin, C. H., Teller, S. J.: Management of Large Amounts of Data in Interactive Building Walkthroughs. In Proc. ACM 0-89791-471-6/92/0003/0011. (1992)
2. Aissi, S., Malu, P., Srinivasan, K.: E-business Process Modeling: The Next Big Step. IEEE Computer. 35 5 (2002) 55-62
3. Chang, B. R.: Hybrid BPNN-Weighted Grey-CLMS Forecasting. Journal of Information Science and Engineering. 21 1 (2005) 209-221
4. Box, G. E. P., Jenkins, G. M., Reinsel, G. C.: Time Series Analysis: Forecasting & Control. Prentice-Hall, New Jersey (1994)
5. Haykin, S.: Neural Network: A Comprehensive Foundation. 2nd Ed. Prentice Hall, New Jersey (1999)
6. Jang, J.-S. R.: ANFIS: Adaptive-Network-based Fuzzy Inference Systems. IEEE Transactions on Systems, Man, and Cybernetics. 23 3 (1993) 665-685
7. Gourieroux, C.: ARCH Models and Financial Applications. Springer-Verlag, New York (1997)
8. Bellerslve, T.: Generalized Autoregressive Conditional Heteroscedasticity. Journal of Econometrics. 31 (1986) 307-327
9. Chang, B. R.: Compensation and Regularization for Improving the Forecasting Accuracy by Adaptive Support Vector Regression. International Journal of Fuzzy System. 7 3 (2005) 109-118
10. Hamilton, J. D.: Time Series Analysis. Princeton University Press, New Jersey (1994)
11. Hentschel, L.: All in the Family: Nesting Symmetric and Asymmetric GARCH Models. Journal of Financial Economics. 39 (1995) 71-104
12. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (1995)

13. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines (and other kernel-based learning methods). Cambridge University Press, London (2000)
14. Kreyszig, E.: Advanced Engineering Mathematics. 8th Edition. Wiley, New York (1999)
15. Chang, B. R.: Forecasting the Flow of Data Packets in Web Using ANFISCH Predictor Tuned by Segmented Adaptive Support Vector Regression. In Proc. The 5th International Conference on Computer and Information Technology. Fudan University, Shanghai, China, Sept. 21-23, (2005) 23-27
16. Chang, B. R.: Applying Nonlinear Generalized Autoregressive Conditional Heteroscedasticity to Compensate ANFIS Outputs Tuned by Adaptive Support Vector Regression. Fuzzy Sets and Systems. 157 13 (2004) 1832-1850
17. Inflow and Outflow of data packets by bits per second in WWW server, computer center, NTTU. http://checknet.nttu.edu.tw:8080/mrtg2/2150.htm
18. Inflow and Outflow of data packets by bits per second in WWW server, computer center, STU. http://dcs.stu.edu.tw
19. Ljung, G. M., Box, G. E. P.: On a Measure of Lack of Fit in Time Series Models. Biometrika. 65 (1978) 67-72
20. Diebold, F. X.: Elements of Forecasting. South-Western, Cincinnati (1998)

A Hybrid Model for Symbolic Interval Time Series Forecasting

André Luis S. Maia, Francisco de A.T. de Carvalho, and Teresa B. Ludermir

Centro de Informatica - CIn/UFPE
Av. Prof. Luiz Freire, s/n - Cidade Universitaria
CEP: 0740-540 Recife-PE, Brazil
{alms3, fatc, tbl}@cin.ufpe.br

Abstract. This paper presents two approaches to symbolic interval time series forecasting. The first approach is based on the autoregressive moving average (ARMA) model and the second is based on a hybrid methodology that combines both ARMA and artificial neural network (ANN) models. In the proposed approaches, two models are respectively fitted to the mid-point and range of the interval values assumed by the symbolic interval time series in the learning set. The forecast of the lower and upper bounds of the interval value of the time series is accomplished through the combination of forecasts from the mid-point and range of the interval values. The evaluation of the proposed models is based on the estimation of the average behaviour of the *mean absolute error* and *mean square error* in the framework of a Monte Carlo experiment.

1 Introduction

For decades, a number of authors have used different statistical methods for modeling and forecasting time series. Such methods vary from a moving average and exponential smoothing to linear and nonlinear regressions. Box and Jenkins [2] developed autoregressive moving average (ARMA) models for time series forecasting. ARMA models are used under linearity presuppositions, that is, the future value of a variable is assumed to be a linear function of several past observations and random errors. However, there are many series for which the linearity supposition is not satisfied. Consequently, ARMA models cannot provide satisfactory results when used to capture the nonlinear structure of data. This leads to an increase in forecasting errors. Various alternative methods have been developed to improve the forecasting of time series with nonlinear patterns, such as the autoregressive conditional heteroscedastic (ARCH) model [6]. Despite such methods demonstrating significant improvements over linear models, they tend to be specific to particular applications.

In classical data analysis, items are usually represented as a vector of quantitative or qualitative measurements for which each column represents a variable. To put it more accurately, each individual takes a single value for each variable. In practice, however, this model is too restrictive to represent complex data. In order to take into account the variability and/or uncertainty inherent to the

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 934–941, 2006.

data, variables must assume sets of categories or intervals, possibly even with frequencies or weights. The aim of *Symbolic Data Analysis* (SDA) [3] is to extend classical data analysis techniques (regression models, clustering, factorial techniques, decision trees, etc.) to what is known as symbolic data [4]. Such data are usually collected in a symbolic data table, where individuals are represented in the rows and variables in the columns. The cells may contain sets of categories, intervals or weight (probability) distributions. In this paper, we address symbolic interval data, i.e., symbolic variables that take an interval as a value.

In the present paper, we first present an extension of the ARMA model estimation methodology for the analysis of symbolic interval data. Next, we introduce a new methodology based on a hybrid of the ARMA and artificial neural networks (ANN), based on a proposal by Zhang [9]. A number of issues led us to consider a hybrid model. Firstly, it is very difficult in practice to determine when a time series is generated by a linear or non-linear process or when one particular method is more efficient than another for forecasting the series. Models are generally adjusted and the one that provides the most accurate results is selected to forecast the series. Nonetheless, due to the influence of other factors (sample variations, uncertainty of the model and changes in the structure of the series), the selected model is not necessarily the best model for predicting the future. Secondly, real time series are rarely purely linear or non-linear processes, and often contain both patterns. Thirdly, in the literature on time series forecasting, there is no single method that is best in all situations. Through the combination of the ARMA and ANN models, complex autocorrelation structures in the data can be modeled so as to obtain greater forecasting accuracy.

In the section 2, we present a brief review of the ARMA and ANN models for time series forecasting. The hybrid model is introduced in the section 3. In the section 4, we demonstrate how we extend these models to handle interval time series. The section 5 describes the framework of Monte Carlo simulations and presents experiments with synthetic interval data sets. Finally, the 6 section offers concluding remarks.

2 Time Series Forecasting Models

The main objectives of a time series analysis include the description of the behaviour of the series, investigation of a plausible mechanism that had generated it and the obtaining of forecasts for its future values [2]. In this section, we introduce the traditional ARMA time series model and *multilayer perceptron* ANN models for time series.

2.1 The ARMA Model

An often-used methodology in handling and predicting time series is known as the Box-Jenkins method, or simply ARIMA. This is method is based on the synthesis of patterns using historical data, transforming knowledge into equations (model) through the estimation of associated parameters. ARMA models are created from a finite, linear combination of past values of the series and/or a finite, linear combination of past errors, for which such errors are the differences between the values predicted by the ARMA model and the real values of the time series. The particular model that will be used in the present paper, known as ARMA(p, q), is represented by the following:

$$y_t = \theta_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \cdots - \theta_q \epsilon_{t-q},$$

where y_t represents the current value of the time series and ϵ_t is the random error at time t; ϕ_i $(i = 1, 2, \ldots, p)$ e θ_j $(j = 0, 1, 2, \ldots, q)$ are the model parameters to be estimated; p and q refer to the order of the model; the random errors ϵ_t are assumed to be independent and identically distributed with a zero average and σ^2 constant variance. Stationarity is a necessary condition in the construction of an ARMA model that is useful for forecasting. The Box-Jenkins methodology includes three iterative steps: model identification, parameter estimation and model diagnosis.

2.2 ANN Model for Time Series

When the linearity restriction regarding model form is relaxed, the number of possible models that can be used in time series forecasting for capturing nonlinear structures is very large. For example, ANN models are able to approximate various forms of non-linearity in the data and, differently from ARMA models, do not require any presupposition regarding the form of the model. The main advantage over other nonlinear models is that ANNs are universal function approximators with a high degree of accuracy [7]. Networks with three layers (one input, one hidden and one output node) connected acyclically are frequently used for modeling and forecasting time series. In the model we use here, the relation between the output, y_t, and inputs, y_{t-1}, y_{t-2},. . ., y_{t-p}, is as follows:

$$y_t = \alpha_0 + \sum_{j=1}^{q} \alpha_j \cdot g\left(\beta_{0j} + \sum_{i=1}^{p} \beta_{ij} y_{t-i}\right) + \epsilon_t,$$

where α_j and β_{ij} are the model parameters; p is the number of input nodes; and q is the number of hidden nodes. Thus, the ANN model performs a nonlinear functional mapping from past observations to the future value in a manner that is equivalent to a nonlinear, autoregressive model. The logistic function is often used as a transference function in the hidden layer [8].

3 The Hybrid Model

ARMA and ANN models have had considerable success in their linear and nonlinear domains, respectively. However, the use of ARMA models for complex non-linear problems may not be adequate. Similarly, using ANNs to model linear problems has produced conflicting results in the literature [9]. Through a combination of different models, different series patterns can be captured. Thus, a hybrid methodology that can simulataneously model linear and nonlinear processes may be a good strategy for practical use. Zhang [9] states that a time series is composed of a linear autocorrelation structure and a nonlinear component,

$$y_t = L_t + N_t, \tag{1}$$

where L_t and N_t respectively denote the linear and nonlinear components. The hybrid model Zhang proposes [9] consists of two steps. First, ARMA is used to model the linear component of the equation (1). The residuals of the ARMA model will contain information on the non-linearity of the series,

$$e_t = y_t - \widehat{L}_t. \tag{2}$$

After adjusting the ARMA model, the residuals are modeled through ANN in order to capture the nonlinear relation of the series using p input nodes.

$$e_t = f(e_{t-1}, e_{t-2}, \dots, e_{t-p}) + \epsilon_t.$$

The prediction of the residual, e_t, is denoted through the ANN model as $\widehat{N}_t$. Thus, the combined forecast provided by the hybrid model will be given by

$$\widehat{y}_t = \widehat{L}_t + \widehat{N}_t.$$

Note that this methodology does not require any presupposition regarding the correlation structure of the time series. For further details on the methodology used, see Zhang [9].

4 Constructing Models for Symbolic Interval Time Series Forecasting

In classical data analysis, each input variable represents a single possible value at an instant in time. The need to consider data that contain information that cannot be represented by classical models has led to the development of SDA. Interval data are those data in which the values of the variables are intervals in $\mathbb{R}$. Different methodologies have been developed to analyze symbolic interval data. One way to represent this type of data is through the mid-point and range of interval [5]. When symbolic interval data are collected in a chronological sequence, we say that we have a time series of symbolic interval data. At each instant in time, $t = 1, 2, \dots, n$, we have X_{U_t} e X_{L_t}, with $X_{L_t} \leq X_{U_t}$ as the upper and lower bounds of the interval, respectively,

$$[X_{L_1}, X_{U_1}], [X_{L_2}, X_{U_2}], \dots, [X_{L_n}, X_{U_n}]. \tag{3}$$

In the method we propose, two time series are considered: (i), the interval mid-point series, X^c; and (ii), the half-range interval series, X^r. Consider the time series (3), we can respectively represent the mid-point and the half-range interval series by (for $t = 1, 2, \dots, n$)

$$X_t^c = \frac{X_{U_t} + X_{L_t}}{2} \quad \text{e} \quad X_t^r = \frac{X_{U_t} - X_{L_t}}{2},$$

4.1 Fitting an ARMA Model for Forecasting Symbolic Interval Time Series Data

The ARMA models for the mid-point series and the half-range interval series that will be used to predict future values of the upper and lower bounds of the intervals are respectively as follows:

$$X_t^c = \alpha_0 + \alpha_1 X_{t-1}^c + \cdots + \alpha_p X_{t-p}^c + \epsilon_t - \beta_1 \epsilon_{t-1} - \cdots - \beta_q \epsilon_{t-q},$$

$$X_t^r = \delta_0 + \delta_1 X_{t-1}^r + \cdots + \delta_p X_{t-p}^r + u_t - \psi_1 u_{t-1} - \cdots - \psi_q u_{t-q}.$$

Note that the parameters of the models are distinct. The values predicted by the ARMA model for the lower and upper bounds of the interval, $\widehat{L}_{U_t}$ and $\widehat{L}_{L_t}$, are respectively given by

$$\widehat{L}_{U_t} = \widehat{X}_t^c + \widehat{X}_t^r \quad \text{e} \quad \widehat{L}_{L_t} = \widehat{X}_t^c - \widehat{X}_t^r,$$

where $\widehat{X}_t^c$ and $\widehat{X}_t^r$ represent the values predicted by the linear adjustment for the mid-point and the half-range interval series.

4.2 Fitting a Hybrid Model for Forecasting Symbolic Interval Time Series Data

The idea here is to use a methodology based on the hybrid system Zhang proposes [9] for modeling the mid-point series and the half-range interval series. According to the equation of the residuals from the linear model (equation 2), we can denote the residuals of the mid-point and half-range adjustments, respectively, as

$$e_{X_t^c} = X_t^c - \widehat{X}_t^c \quad \text{e} \quad e_{X_t^r} = X_t^r - \widehat{X}_t^r,$$

where $\widehat{X}_t^c$ and $\widehat{X}_t^r$ are the values predicted by the linear adjustment for the mid-point and half-range inteval series; and X_t^c and X_t^r are the corresponding observed values. Thus, after the linear adjustment of the interval series, we have two new series: one from the nonlinear residuals of the adjustment of the interval mid-point series, $e_{X_t^c}$; and the other from the nonlinear residuals of the adjustment of the half-range interval series, $e_{X_t^r}$.

It should be pointed out that in this proposal there is no need for strict pressupositions regarding the model (linearity, same correlation structure for the interval bounds of the series, etc.). Thus, the final forecast of the interval time series bounds is given by

$$\widehat{y}_{U_t} = \widehat{L}_{U_t} + \widehat{N}_{U_t} \quad \text{e} \quad \widehat{y}_{L_t} = \widehat{L}_{L_t} + \widehat{N}_{L_t},$$

where $\widehat{N}_{U_t}$ and $\widehat{N}_{L_t}$ are the errors predicted by ANN for the upper and lower bounds of the interval, respectively, obtained from the following:

$$\widehat{N}_{U_t} = \widehat{e}_{X_t^c} + \widehat{e}_{X_t^r} \quad \text{e} \quad \widehat{N}_{L_t} = \widehat{e}_{X_t^c} - \widehat{e}_{X_t^r},$$

where $\widehat{e}_{X_t^c}$ and $\widehat{e}_{X_t^r}$ respectively represent the error for the mid-point series and the error for the half-range interval series at time t.

5 Empirical Results

In this section, we will demonstrate the efficiency of the proposed models through experiments with symbolic interval time series data simulated with different degrees of difficulties.

In order to assess the performance of the ARMA model and the hybrid model in terms of accuracy in the adjustment and forecasting of interval time series, we have simulated some series with 200 observations. The simulations of the interval time series were executed as follows:

1. Let time series X_t^c $(t = 1, 2, \ldots, n)$, which represents the interval mid-point series $X_t = [X_{L_t}, X_{U_t}]$ $(t = 1, 2, \ldots, n)$, be a process generated with a known structure, such as an AR(1) process;
2. After obtaining the mid-point series, we construct the half-range interval series. Suppose time series X_t^r $(t = 1, 2, \ldots, n)$, which represents the half-range interval series $X_t = [X_{L_t}, X_{U_t}]$ $(t = 1, 2, \ldots, n)$, is uniformly distributed in the interval $[a, b]$, for example, $X_t^r \sim U[10, 20]$;
3. In the generation of the interval time series, we know that the series $X_t = [X_{L_t}, X_{U_t}]$ presents a relation to X_t^c e X_t^r as follows: $X_{L_t} = X_t^c - X_t^r$ e $X_{U_t} = X_t^c + X_t^r$;
4. At each iteration, the simulated symbolic interval time series is partitioned in a learning set (188 observations) and test set (12 observations).

Table 1 shows four different configurations for the generation of the interval time series that were used to compare the performances of the ARMA and hybrid models in distinct situations. These configurations consist of a combination of the mid-point and range series generated. The first two configurations, C_1 and C_2, present a linear relation between the future value and past value of the mid-point series plus a random error term, ϵ_t, normally distributed with a zero average and constant variance, $\epsilon_t \sim N(0, 1)$. The nonlinear configurations C_3 and C_4 are examples that present complex, chaotic behaviour. It is expected that the use of the ARMA models for complex nonlinear problems does not lead to satisfactory forecasting results, and therefore, the hybrid model provides greater forecasting accuracy.

Table 1. Simulated symbolic interval time series configurations

Configuration	X^c process	X^r process
C_1	$X_t^c = 10 + 0.7X_{t-1}^c + \epsilon_t$	$U[10, 12]$
C_2	$X_t^c = 1 + X_{t-1}^c + \epsilon_t$	$U[8, 10]$
C_3	$X_t^c = 4X_{t-1}^c(1 - X_{t-1}^c)$	$U[2, 5]$
C_4	$X_t^c = 0.2\frac{X_{t-17}^c}{1+(X_{t-17}^c)^{10}} - 0.1X_{t-1}^c$	$U[2, 4]$

5.1 Experimental Evaluation

The performance evaluation of the proposed interval time series forecasting models, ARMA and a hybrid model, was accomplished through the following measures: upper bound mean absolute error (MAD_U), lower bound mean absolute error (MAD_L), upper bound mean square error (MSE_U) and lower bound mean square error (MSE_L). The values of the error measures were obtained from the observed values $X_t = [X_{L_t}, X_{U_t}]$ ($t = 1, 2, \ldots, n$) and corresponding predictive values $\widehat{X}_t = [\widehat{X}_{L_t}, \widehat{X}_{U_t}]$.

The forecasting error measures were calculated for the ARMA model and the hybrid model in the framework of a Monte Carlo experiment with 100 iterations in the test set (12 observations). At the end of the experiments, the average and standard deviation were calculated for the MAD_U, MAD_L, MSE_U and MSE_L measures in the 100 Monte Carlo iterations.

The selection of the best ARMA model for adjusting the mid-point and range of interval series was accomplished through the minimization procedure of the *Akaike Information Criterion* (AIC) (ser Akaike [1]). The parameters were estimated for maximum likelihood. Multilayer perceptron networks with three layers of units (one input, one hidden and one output units) connected acyclically trained with the backpropagation algorithm are used in the hybrid model.

Table 2 displays the results of the Monte Carlo experiments for the four configurations. The standard deviations for the error measures considered are in parentheses. We can see that the hybrid model presented a higher average performance than the ARMA model in forecasting interval time series for all the situations considered. Even for the series with a linear correlation structure, (C_1 and C_2), the hybrid model increased the accuracy of the predictions. This superiority is better

Table 2. Average and standard deviation of the mean square errors and mean absolute errors calculated from the 100 replications in the framework of a Monte Carlo experiment for the test set

Configuration C_1				
	MAD		MSE	
Model	X_U	X_L	X_U	X_L
ARMA	0.881 (0.082)	0.901 (0.081)	1.249 (0.191)	1.290 (0.191)
Hybrid	0.742 (0.081)	0.775 (0.088)	0.983 (0.187)	1.047 (0.197)
Configuration C_2				
	MAD		MSE	
Model	X_U	X_L	X_U	X_L
ARMA	0.870 (0.079)	0.868 (0.081)	1.048 (0.178)	1.047 (0.181)
Hybrid	0.737 (0.093)	0.748 (0.111)	0.757 (0.167)	0.757 (0.194)
Configuration C_3				
	MAD		MSE	
Model	X_U	X_L	X_U	X_L
ARMA	0.444 (0.089)	0.451 (0.089)	0.289 (0.099)	0.298 (0.098)
Hybrid	0.354 (0.070)	0.362 (0.066)	0.189 (0.063)	0.192 (0.058)
Configuration C_4				
	MAD		MSE	
Model	X_U	X_L	X_U	X_L
ARMA	0.291 (0.057)	0.289 (0.061)	0.125 (0.041)	0.122 (0.042)
Hybrid	0.226 (0.034)	0.224 (0.035)	0.074 (0.020)	0.073 (0.018)

observed in the nonlinear configurations(C_3 and C_4). No substantial differences were observed between the two methods regarding standard deviations.

6 Concluding Remarks

In the present paper, we present two new methods for modeling and forecasting symbolic interval time series. The first is based on the autoregressive moving average model (ARMA), and the second is a hybrid model using both ARMA and artificial neural networks. In the proposed methods, we adjusted the models in the mid-point and interval range series of the training set. The prediction of the future values of the lower and upper bounds of the intervals was accomplished through the combination of the mid-point and interval range forecasts.

The evaluation of the proposed methods was accomplished through the average behaviour of the mean absolute error and the mean square error of the forecasts in the framework of a Monte Carlo experiment. The Monte Carlo simulations demonstrated that both methods present a satisfactory performance in forecasting interval series with either a linear or nonlinear behaviour. However, the hybrid model that uses ARMA to model the linear component of the series and artificial neural networks to capture the nonlinear relation presented a better average performance with regard to the error measures considered. We noted that the hybrid model outperformed the ARMA model even in situations where the series had a linear behaviour. When the series presented chaotic behaviour, the hybrid model was far superior to the ARMA model.

Acknowledgments. The authors would like to thanks CNPq (Brazilian Agency) for its financial support.

References

1. H. Akaike, A new look at the statistical model indentification, *IEEE Transactions on Automatic Control*, **19**, 716–723, 1974.
2. G. E. P. Box and G. M. Jenkins, *Time Series Analysis, Forecasting and Control.* San Francisco: Holden Day, 1976.
3. H. H. Bock and E. Diday, *Analysis of Symbolic Data*, Springer, Berlin Heidelberg, 2000.
4. L. Billard and E. Diday, *Symbolic regression Analysis.* Proceedings of the 8th Conference of the International Federation of Classification Societies, IFCS-2002, Crakow (Poland), Jajuga, K. et al. Eds, Springer, 281–288, 2002.
5. De Carvalho, F.A.T, Lima Neto, E.A., Tenorio, C.P.: A New Method to Fit a Linear Regression Model for Interval-Valued Data. Lecture Notes on Artificial Inteligence, LNAI 3238, KI-04, Ulm (Germany), Biundo, S. et al. Eds,(2004), Springer, 295-306.
6. R. F. Engle, Autoregressive conditional heteroscedasticity with estimates of the variance of UK inflation, *Econometrica*, **50**, 987–1008, 1982.
7. K. Hornik, M. Stinchcombe and H. White, Multilayer feedforward networks are universal approximators, *Neural Networks*, **2**, 359–366, 1989.
8. Z. Tang and P. A. Fishwick, Feedforward neural nets as models for time series forecasting *ORSA Journal of Computing.* **5**, 374–385, 1993.
9. G. Zhang, Time Series forecasting using a hybrid ARIMA and neural network model, *Journal of Neurocomputing*, **50**, 159–175, 2003.

Peak Ground Velocity Evaluation by Artificial Neural Network for West America Region

Ben-yu Liu, Liao-yuan Ye, Mei-ling Xiao, and Sheng Miao

Institute of Public Safety and Disaster Prevention, Yunnan University,
650091 Kunming, China
{liubenyu, lyye, mlxiao,
msheng}@ynu.edu.cn
http://www.srees.ynu.edu.cn/structure/index.html

Abstract. With the Peak Ground Velocity 283 records in three dimensions, the velocity attenuation relationship with distance was discussed by neural network in this paper. The earthquake magnitude, epicenter distance, site intensity and site condition were considered as basic input element for the network. By using Bayesian Regularization Back Propagation Neural Networks (BRBPNN), the over-fitting phenomenon was reduced to some extent. The horizontal velocity was discussed. The PGV predicted by neural networks can simulate the detail difference with distance, while the PGV given by other traditional attenuation relationship only give a reduction relation with distance. The importance of each input factor was compared by the square weight of the input layer of the network. The order may be earthquake magnitude, epicenter distance and soil condition.

1 Introduction

The seismic parameters attenuation relationship is very important for engineers, since it was always used to predict the ground movement of a scenario earthquake. Traditionally, there are two ways to get this special relationship. One is make a theoretical attenuation model and analyze its influence coefficients and parameters; the other is statistical way from the stations seismic records. Since the complicated nature of the problem, the predicted results of most relationships can not accord with the stations records nice.

As we know, Artificial Neural Networks (ANN) are highly parametric functions of the input variables through processing units, whose high connectivity makes them suitable for describing complex input-output mappings without resorting to a physical description of the phenomenon. Some studies in this problem have been reported. Zheng Guanfen discussed the earthquake intensity attenuation using Back Propagation Neural Networks (BPNN) [1]. Wang Hushuang constructed a NN model to simulate the peak seismic parameters attenuation relation and a NN to relate the intensity with peak seismic parameters [2]. By using the peak horizontal acceleration records acquired in Lancang-Gengma Earthquake in 1988, Cui Jianwen constructed a NN model to predict the peak ground acceleration attention relationship in Yunnan region [3].

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 942 – 951, 2006.

Unfortunately, most work on attenuation relationship just discussed on Peak Ground Acceleration. Theoretically speaking, the velocity can be integrated from acceleration. Since the complex nature of this problem, the result from this way cannot provide a reasonable solution. Based on the data records, the horizontal peak ground velocity was discussed by the neural networks.

2 Experimental Statistical Model

The experimental models were simply regressed from the strong earthquakes records. As for the records in West America, the following models were famous.

1978, McGuire [4] (West America, $j_s = 0$ is rock, $j_s = 1$ is soil)

$$\ln v = -1.00 + 1.07M - 0.96 \ln R - 0.07 j_s, \quad \sigma_{\ln v} = 0.64 . \tag{1}$$

1981, Joyner-Boore [5] (West America, $j_s = 0$ is rock, $j_s = 1$ is soil)

$$\ln v = -1.542 + 1.13M - \ln R - 0.0059R + 0.39 j_s, \quad \sigma_{\ln v} = 0.51 . \tag{2}$$

$$R = \sqrt{D^2 + 4^2} . \tag{3}$$

1979, Espninosa [4] (West America, $M = 4.0 \sim 7.2$)

$$v = 6.17 \times 10^{-4} e^{2.3M} \Delta^{-1.35} . \tag{4}$$

This model did not include the magnitude saturation which is proved in many strong earthquakes. Huo Junrong [6] (West America, rock) revised this model in order to consider the mag-nitude saturation, as following,

$$\lg v = -1.148 + 0.8241M - 1.7941 \lg(D + 0.3268 e^{0.6135M}), \quad \sigma_{\lg v} = 0.2582 . \tag{5}$$

Where v is peak ground velocity, R is the distance from the observation site to the focal, D is the direct distance from the observation site to the striking fault and M is the Richter Magnitude of the earthquake.

The standard error of Huo's is similar with other formulae. Because $\ln 10 = 2.3026$, thus the $\sigma_{\lg v} = 0.2582$ is relevant to $\sigma_{\ln v} = \ln 10 \times \sigma_{\lg v} = 0.5945$.

3 Neural Network Model for Peak Ground Velocity Attenuation

The main characteristics of neural networks are their ability to learn nonlinear functional relationships from examples and to discover patterns or regularities in data through self-organization. The neural network learning process primarily involves the iterative modification of the connection weights until the error between the predicted and expected output values is minimized. It is through the presentation of examples, or training cases, and application of the learning rule that the neural network obtains the relationship embedded in the data.

3.1 Back Propagation Neural Networks Design

It is nature that the neural network designed for this problem should be accordance with the sample data. Thus, the input vector of the network contains 4 components. While for the output vector, it includes only one component.

As for back propagation neural network, it has been proved the only two layers can achieve arbitrary nonlinear mapping if the neurons in the hidden layers are not limited. Thus, a two layers neural network is constructed for the problem.

Thus, the parameters in Figure 1 can be decided as $R=4$, $S^1=20$, $S^2=1$.

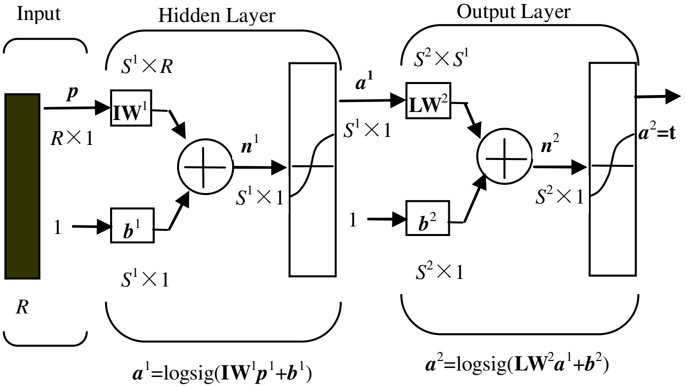


Fig. 1. Two layers neural works for soil liquefaction prediction

3.2 Theory on the Bayesian Regularization Back Propagation Neural Networks

We divided the datasets into two parts: training and testing. In using multiply layer propagation network, the problem of over-fitting on noise data is of major concern in network design strategy. The initial results of using a standard BP algorithm showed poor generalization performance and slow speed of training. To overcome these shortcomings, we incorporated Bayesian learning to this work. In the Bayesian framework, a weight decay term is introduced to the cost function (or performance index) given by

$$F(w) = \alpha E_W + \beta E_D . \tag{6}$$

where E_W is the sum square of the networks weights, E_D is the sum square of the error between network outputs and targets, α and β are hyper-parameters for the target function. The relative value of α and β determined the emphasis on the network training on minimization of the output errors or the volume of the network. As shown in Equation (6), the main problem with implementing regularization is to set/learn the correct values for the parameters in the cost function. Ref. [7-9] has presented extensive works on the application of Bayesian rule to neural network training and to optimizing regularization.

In the Bayesian framework, the weights of the network are considered the random variables. The weights in the network are adjusted to the most probable weight parameter, w_{MP}, given the data $D\{(x^m, t^m)\}$, network configuration (M_i), and hyper-parameters, i.e., α and β.

Set the α and β as stochastic variables, the Bayesian rule is used for evaluating the posterior probability of α and β. This is given by

$$P(\alpha,\beta \mid D,M_i) = \left(P(D \mid \alpha,\beta,M_i)P(\alpha,\beta \mid M_i)\right)/P(D \mid M_i). \tag{7}$$

where $P(\alpha,\beta \mid M_i)$ represents the prior probability of the hyper-parameters and are generally chosen to be uniformly distributed. Since $P(D \mid M_i)$ is independent of α and β, maximum posterior values for hyper-parameters can be found by maximizing the likelihood term $P(D \mid \alpha,\beta,M_i)$.

Using Bayesian rule, the posterior probability of the weight parameters is:

$$P(w \mid D,\alpha,\beta,M_i) = \frac{\left(P(D \mid w,\beta,M_i)P(w \mid a,M_i)\right)}{P(D \mid \alpha,\beta,M_i)}. \tag{8}$$

Assume the error and the weight is distributed in Gaussian form,

$$P(D \mid w,\beta,M_i) = \exp(-\beta E_D)/Z_D(\beta). \tag{9}$$

$$P(w \mid \alpha,M_i) = \exp(-\alpha E_W)/Z_W(\alpha). \tag{10}$$

If the $P(D \mid \alpha,\beta,M_i)$ in Equation (8) is regularized factor, then $P(w \mid D,\alpha,\beta,M_i)$ must equal to $\exp(-F(w))/Z_F(\alpha,\beta)$. Take them into Equation (7),

$$P(D \mid \alpha,\beta,M_i) = Z_F(\alpha,\beta)/(Z_W(\alpha)Z_D(\beta)). \tag{11}$$

where

$$Z_W(\alpha) = (2\pi/\alpha)^{N/2}. \tag{12}$$

$$Z_D(\beta) = (2\pi/\beta)^{N/2}. \tag{13}$$

$$Z_F(\alpha,\beta) \approx \exp(-F(w_{MP}))(2\pi)^{N/2}|A|^{-1/2}. \tag{14}$$

where $A = \beta\nabla^2 E_D + \alpha\nabla^2 E_W$ is the Hessian matrix of the target function F. Further, the log the Equation (12), then differentiating it with respect to α and β, and setting it to zero, the optimal values of α and β can be obtained by

$$\alpha_{MP} = \gamma / \left(2E_W\left(w_{MP}\right)\right). \tag{15}$$

$$\beta_{MP} = (n-\gamma)/\left(2E_D\left(w_{MP}\right)\right). \tag{16}$$

$$\gamma = N - \alpha_{MP}\mathrm{trace}^{-1}\left(A_{MP}\right). \tag{17}$$

where n is the number of sample, N is the number of parameter in the network, γ is the number of effective parameters which may reduce the error function for the network in training process.

3.3 Training and Testing for BRBPNN Model

For the earthquake movement records in west America region, four elements were considered as the factors for seismic movement attenuation relationship. They are earthquake magnitude, epicenter distance, site intensity and site condition. The records were in three directions, two records were in horizontal direction and one was vertical direction. 236 USGS stations were found PGV records in 69 earthquakes. Part of data is listed in Table 1.

Table 1. Parts of training and testing records samples [4]

USGS No.	Magnitude	Distance (km)	Intensity	Site	Direction	PGV(cm/s)
117	7.1	9.3	8	S	S00E	33.45
117	7.1	9.3	8	S	S90W	36.92
117	7.1	9.3	8	S	VERT	10.84
1023	6.0	53.0	5	S	S44W	4.81
1023	6.0	53.0	5	S	N46W	7.39
1023	6.0	53.0	5	S	VERT	2.21
475	7.7	109.0	7	S	S00E	6.23
475	7.7	109.0	7	S	S90W	9.07
475	7.7	109.0	7	S	VERT	4.53
1095	7.7	43.0	7	S	N21E	15.72
1095	7.7	43.0	7	S	S69E	17.71
1095	7.7	43.0	7	S	VERT	6.67

Taking the horizontal records out from these data, thus 472 records were found in two horizontal directions. Four fifth of the records was taken as the training samples, and the others as testing samples.

The relevant errors of the result of the neural network with the original observed data were showed in Figure 2. In order to demonstrated the efficiency of the neural networks, the errors predicted by Joyner [5] was showed in Figure 3 too.

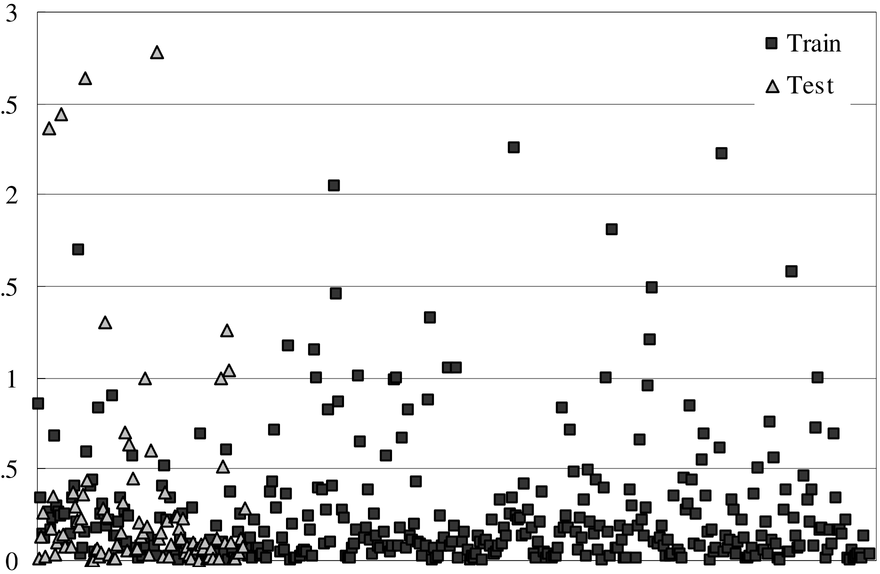


Fig. 2. The errors of observed PGV with that of neural networks

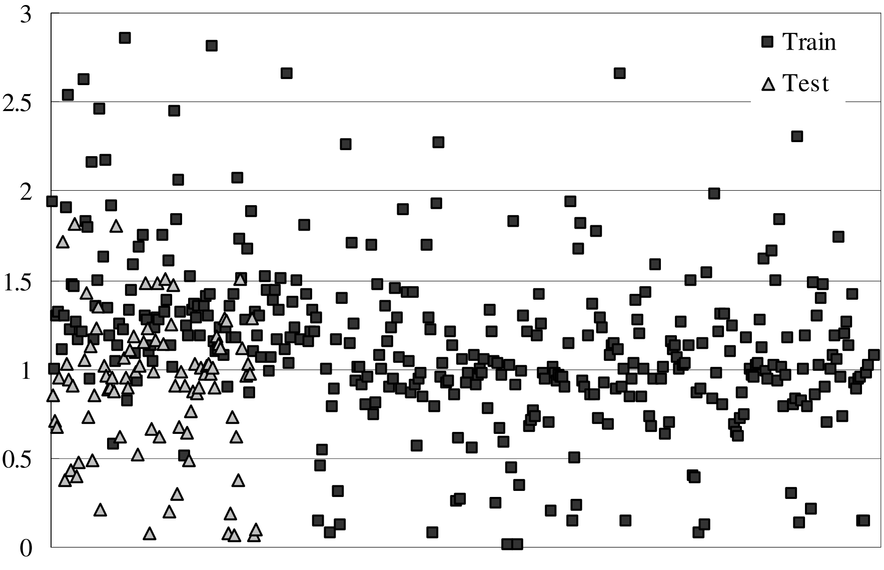


Fig. 3. The errors of observed PGV with that of Joyner-Boore [5]

It is easy to see that errors of neural networks were centered on the baseline; while errors of Joyner [5] were centered on 1. The detail of errors statistic listed in Table 2.It can be see in Table 2 that the neural network model can simulate the records with rather good satisfaction.

Table 2. Errors statistic of neural network model with Joyner Boore model

	0<error<1	1<error<2	2<error<3	error>3	SUM
Neural work	444	15	7	6	472
	94.1%	3.1%	1.5%	1.3%	100%
Joyner Boore	195	248	15	14	472
	41.3%	52.5%	3.2%	3.0%	100%

3.4 Peak Ground Velocity Prediction

The model can be used to predict the peak ground velocity at certain magnitude and certain distance. Since the data in the west America were abundant in M6.6 and M6.9, these two magnitudes were discussed. In order to demonstrate the character of neural network, the model of Joyner-Boore is drawn in Figure 4 with the distance change from 5 to 200 kilometers.

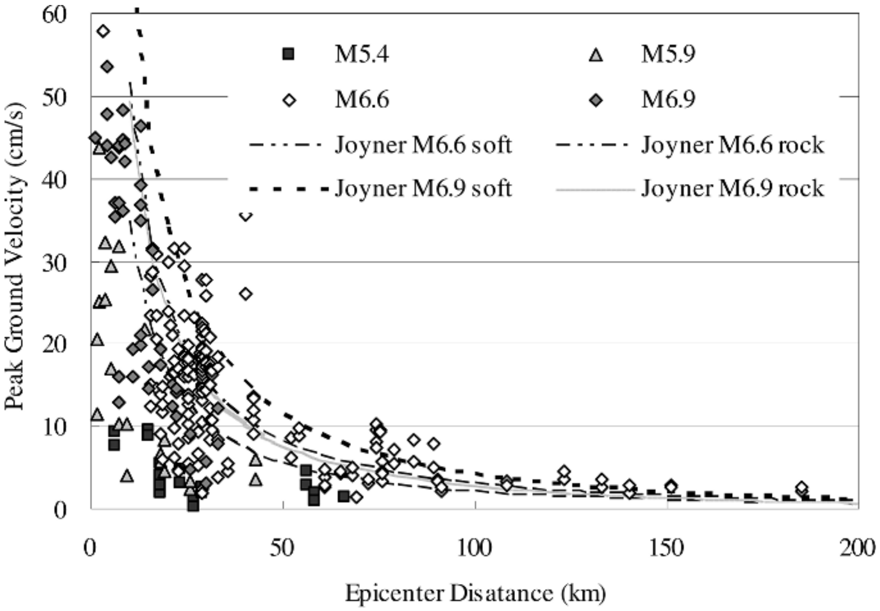


Fig. 4. The attenuation relationship of peak ground velocity predicted by Joyner-Boore, with the site condition of soft soil and rock base. The records of west America on M5.4, M5.9, M6.6 and M6.9 were also pointed.

The relationships predicted by neural network with the original data were showed in Figure 5. It can easily by found that the relationship predicted by traditional regression method such as Joyner-Boore just give a relative smooth line; while the relationship of neural networks can give a up and down line to describe the complicate nature of the PGV attenuation problem.

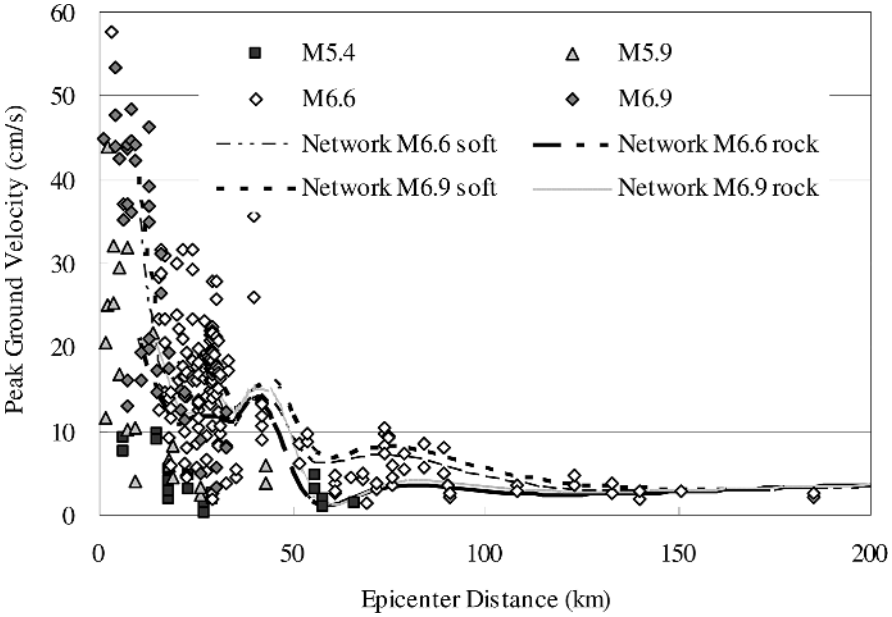


Fig. 5. The attenuation relationship of peak ground velocity predicted by Neural Network, with the site condition of soft soil and rock base. The records of west America on M5.4, M5.9, M6.6 and M6.9 were also pointed.

The importance of each input factor was determined by the square weight of the input layer of the network. Two networks were calculated. The M4 include all the parameters such as magnitude, epicenter distance, site intensity and soil condition. The M3 model just delete the site intensity, for which may indicate the ground movement to some extent. The percent of each input factor was listed in Table 3.

Table 3. The percent of the square of weight for the input layer for each factor

Model	Magnitude	Distance (km)	Intensity	Site condition
M4	35.6437	6.657	20.119	37.5802
M3	50.338	35.6095		14.0526

From the square of the weight in the first layer, the importance order of these factors is magnitude, epicenter distance and soil condition. The percent of the square of weight for the input layer for each factor were showed in Figure 6.

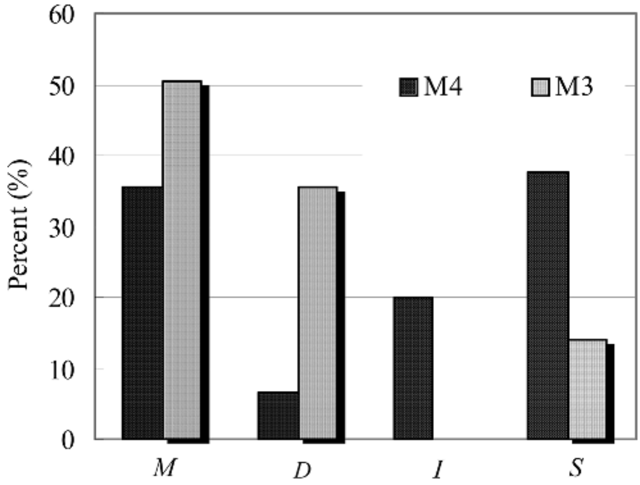


Fig. 6. The percent of the square of weight for the input layer for each factor for M4 and M3

4 Conclusions and Suggestions

From the point of spastics, this paper presented a Bayesian Regularization Back Propagation Neural Networks model to predict the peak ground velocity based on the horizontal records of the West America. The errors trained and tested by neural network model were center on 0; while the errors of Joyner-Boore were centered on 1. This indicated that the neural network model can simulate the observed data with rather good satisfaction. For the certain magnitude 6.6 and 6.9, the neural network model showed an up and down lines for soil and rock condition, which demonstrated the character of neural network model.

The PGV predicted by neural networks can simulate the detail difference with distance, while the PGV given by other traditional attenuation relationship only give a reduction relation with distance. The importance of each input factor was compared by the square weight of the input layer of the network. The order may be earthquake magnitude, epicenter distance and soil condition.

The Neural Networks can be used as a good substitution in engineering application with the accumulation of ground earthquake records.

Acknowledgements

This research is financial supported by the National Science Foundation of China under the grant number 40564002, the National Science Foundation of Yunnan Province under the grant number 2004E007R and the Science and Engineering Foundation of Yunnan University under the grant number 2004Z005B.

References

1. Zhen, G.F., Tao, X.X.: Construct the Intensity Attenuation Relation Using ANN Method. Earthquake Engineering and Engineering Vibration 13(1) (1993) 60–66
2. Wang, H.S..: Intelligent Prediction of the Peak Seismic Parameters Based On ANN. Journal of Seismology 15(2) (1993) 208–216
3. Cui, J.W., Fan, Y.X., Wen R.Z.: Establishment of Attenuation Law of Acceleration Peak Value by Using Neural Network. Earthquake Research 20(3) (1997) 296–306
4. Hu, Y.X., Zhang, Y.M., Shi, Z.L.: Training Material for the Code of Evaluation of Seismic Safety for Engineering Sites. Engineering Earthquake Research Center (1994)
5. Joyner, W.B., Boore, D.M.: Peak Attenuation and Velocity from Strong-Motion Records Including Records from 1979 Imperial Valley, California, Earthquake. Bulletin of Seismology Society of America 71(6) (1981) 2011–2038
6. Huo, J.R.: Near Field Ground Motion Attenuation Research. Ph. D. Thesis of the Institute Of Engineering Mechanics, China Earthquake Administration (1989)
7. MacKay, D. J. C.: Bayesian Interpolation. Neural Computation 4(3) (1992) 415–447
8. MacKay, D. J. C.: A practical Bayesian framework for back propagation networks. Neural Computation 4(3) (1992) 448–472
9. Foresee F. D., Hagan M.T.: Gauss-Newton approximation to Bayesian regularization. Proceedings of the 1997 International Joint Conference on Neural Networks (1997) 1930–1935

Forecasting Electricity Demand by Hybrid Machine Learning Model

Shu Fan[1], Chengxiong Mao[2], Jiadong Zhang[1], and Luonan Chen[1]

[1] Osaka Sangyo University, Daito, Osaka 574-8530, Japan
fanshu@hotmail.com,
tyoukatou@hotmail.com,
chen@elec.osaka-sandai.ac.jp
[2] Huazhong University of Science and Technology, Wuhan 430074, China
cxmao@mail.hust.edu.cn

Abstract. This paper proposes a hybrid machine learning model for electricity demand forecasting, based on Bayesian Clustering by Dynamics (BCD) and Support Vector Machine (SVM). In the proposed model, a BCD classifier is firstly applied to cluster the input data set into several subsets by the dynamics of load series in an unsupervised manner, and then, groups of 24 SVMs for the next day's electricity demand curve are used to fit the training data of each subset. In the numerical experiment, the proposed model has been trained and tested on the data of the historical load from New York City.

1 Introduction

Load forecasting has always been an essential instrument in power system planning and operation. Many operating decisions are based on load forecasts, such as unit commitment, dispatch scheduling of generating capacity and reliability analysis, etc. However, the electric load is increasingly becoming difficult to forecast because of the variability and non-stationarity of load series that result from the dynamic bidding strategies of market players and price-dependent loads as well as time-varying prices. Therefore, more sophisticated forecasting tools with a higher accuracy are necessary for modern power system.

A wide variety of techniques have been tried for load forecasting during the past years [1], most of which are based on time series analysis. Statistical models are firstly adopted for the load forecasting problem, which include linear regression models, data mining approach, autoregressive and moving averages (ARMA) models and Box-Jenkins methods [2]-[4]. Basically, most of the statistical methods are based on linear analysis. However, the load series are usually nonlinear functions of the exogenous variables. Therefore, to incorporate the nonlinearity, the artificial neural networks (ANNs) have received much more attentions recently [1], [5]-[8]. ANNs based methods report a fairly good performances in load forecasting. Recently, a new approach based on machine learning techniques and Support Vector Machines (SVM) has also been used for load forecasting or classification and achieved good performances [9], [10].

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 952 – 963, 2006.

The purpose of this paper is to apply the new advances in machine learning technique to develop a novel and effective load forecasting model. The proposed model adopts a hybrid architecture based on an integration of two machine learning techniques: Bayesian Clustering by Dynamics (BCD) and SVM (or exactly SVR (Support Vector Regression)). Its working procedure can be stated as follows: firstly, a BCD classifier is used to identify the switching or piece-wise stationary dynamics for the load series and to partition the input dataset into several subsets, so that the dynamics in each subset are similar; then groups of 24 SVRs are applied to respectively fit the hourly electricity load data in each partitioned subset by taking advantage of all past information and similar dynamic properties (e.g. piece-wise stationarity). After being trained, the forecasting model can predict the next-day electricity load with an high level of accuracy on the specific subset in a ‘first past the post’ voting manner among the BCD and SVRs, where the output of only one SVR model is used for the final forecast.

Benefited from the application of the hybrid architecture, the proposed model has the following characteristics or advantages:

1) It can handle the non-stationarity of load series well. The hybrid architecture is well suited to capture the dynamics of electricity load time series. The BCD classifier models the process generating each load series as an autoregressive model of order *p*, say AR(*p*), and then clusters those load time series with a high posterior probability of being generated by the same AR(*p*) model [11], [12]. BCD is based on unsupervised learning, which has the ability to partition the space of input training data set into many subsets without prior knowledge about the classifying criteria.

2) It can fit the data well due to multiple local models. Previous works have shown that, the characteristics of load series between regular workdays and anomalous days are quite different [7], [8]. To achieve good forecasting results, the regular workdays and anomalous days should be treated with different schemes. Therefore, we use different feeders of SVRs for the regular weekdays and anomalous days, which means the network can adapt to different models automatically and improve the forecasting accuracy for the anomalous days.

In the experiment, we adopt the load data of New York City to verify the effectiveness of the learning and forecasting for model. For comparative study, we also examine the model only with SVRs.

2 Task Descriptions and Load Data Analysis

This paper uses the hourly electrical demand series of New York City as a test example of our method by comparing with the prediction of New York Independent System Operator (ISO) [13]. In this section, we examine the main characteristics of the hourly load series. Firstly, according to the electricity demand series of New York City from January, 2001 to December, 2003, it can be concluded that there exist different regimes in the load time series due to market and season effects, which generally give rise to piece wise stationary dynamics.

It is well known that temperature information is very important for load forecasting. So it is necessary to analyze the inherent correlation between load and temperature for a specific system. The correlation between the load and temperature can be shown in Fig.1. According to Fig.1, there exists approximately a piecewise linear relationship of correlations between load and temperature with about 50's degree difference of their tangents. The correlation in each segment can be computed using the following expression.

$$\rho_{t,d} = \frac{Cov(t,d)}{\sigma_t \sigma_d} \tag{1}$$

where *Cov(t,d)* is the covariance of temperature *t* and load *d*, and σ_t and σ_d are the standard deviations for *t* and *d*.

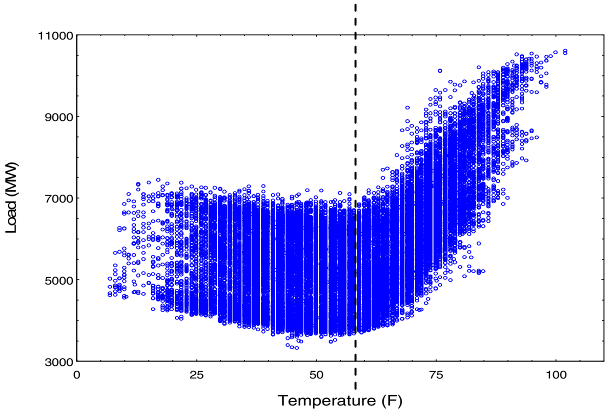


Fig. 1. Correlation between demand and temperature from Jan.1, 2001 to Dec.31, 2003

The dotted line in Fig.1 indicates the separate point between the two piecewise segments, which is obtained by maximizing the absolute values of the two correlation coefficients on both segments. According to our computation, the separating point is approximately 54.5 degrees, and $\rho_{t,d}$ on the two segments is -0.20 and 0.68 respectively. This information will be used in the modeling of the load series in the next section.

3 Method and the Learning Algorithm

3.1 The Architecture of the Forecasting System

In this paper, a time series based nonlinear discrete-time dynamical model, is represented by (2) for the load forecasting,

$$y(t+1) = f(y(t),..., y(t-m+1);T) \tag{2}$$

where *y(t)* is a vector representing the daily electricity load profile at time t, and m is the orders of the dynamical system, which is a predetermined constant. *T* is a vector representing the control parameters of the dynamical system, such as temperature, humidity and day types.

An integrated machine learning model is proposed to reconstruct the dynamics of electric load using the time series of its observables. The forecasting system is shown in Fig. 2.

In Fig. 2, the input variables are different for the BCD and SVR networks, which are given in Tables 1, 2 and 3, respectively.

Table 1. List of input data of the BCD classifier

Input	Variable	Detail description
1-10	Load series L_0	Average daily load series of previous ten days
11	Calendar info *C*	Distinguish anomalous days
12-13	Variables to determine prior probability	*T*: Temperature sensitivity coefficient *H*: Forecasted next day's maximal humidity

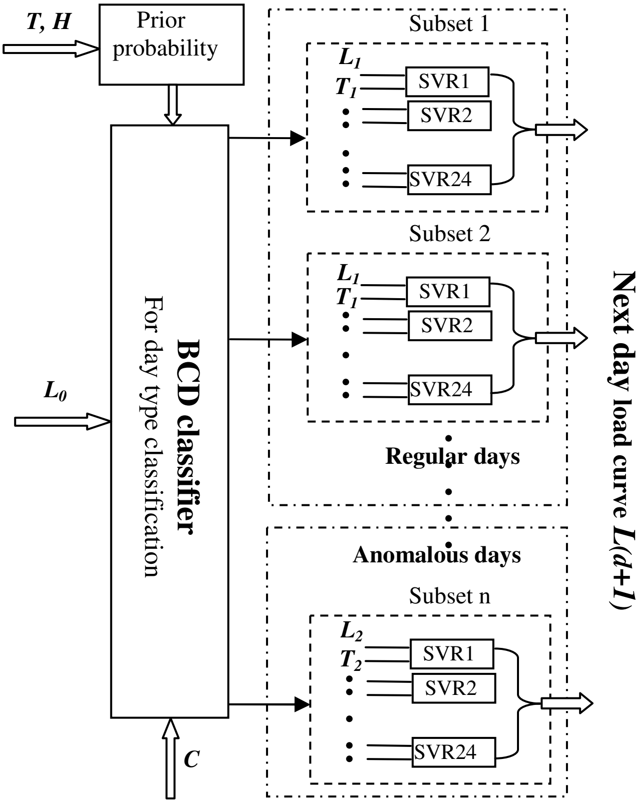


Fig. 2. Integrated machine learning model for the electricity load forecasting

In Table 1, the temperature sensitivity coefficient represents the different correlation between load and temperature. If the forecasted next day's maximal temperature is larger than 54.5 degrees, this coefficient is set as 1, otherwise 0.

For the SVR, in addition to the forecasted and actual temperature, the input variables are the hourly load values of the last day available and the similar hours in the previous days or weeks. As mentioned above, it is necessary to use different feeders for the regular days and anomalous days. The input data of the SVR network for regular days is shown in Table 2.

Table 2. List of input data of the SVR network for regular days

Input	Variable name	Lagged value (hours)
1-9	Hourly load (L_1)	24,25,26,48,72,96,120, 144,168
10-19	Hourly temperature (T_1)	0,1,2,24,48,72,96,120, 144,168

Assuming that the hour of load predication is at 0, the lag 0 represents the target instant.

According to the historical load data, the same type of holiday showed a similar trend of load profile as in previous years. For instance, several studies conclude that the load diagram of holiday has strong connection with that of the two Saturdays before that day and the most recent diagram available, and holidays' forecasts should be assessed as a function of weekend behavior. Based on this analysis, the input data of the SVR network for anomalous days are selected and shown in Table 3.

Table 3. List of input data of the SVR network for anomalous days

Input	Variable name		Lagged hours
1-9	Hourly load	L_2	24,25,26,48,72
	Hourly load of the previous Saturday		h,h-1*
	Hourly load of the previous Sunday		h,h-1*
10-19	Hourly temperature	T_2	0,1,2,24,48,72
	Hourly load of the previous Saturday		h,h-1*
	Hourly load of the previous Sunday		h,h-1*

* h stand for the same clock with the target hour.

3.2 Learning Algorithm: The BCD Classifier

The clustering method implemented in BCD is based on a novel concept of similarity for time series: two time series are similar when they are generated by the same stochastic process. Therefore, the components of BCD are a model describing the dynamics of time series, a metric to decide when two time series are generated by the same stochastic process, and a search procedure to efficiently explore the space of possible clustering models.

BCD models time series by autoregressive equations [12]. Let $S_j = \{x_{j1}, \ldots, x_{jt}, \ldots, x_{jn}\}$ denote a stationary time series of continuous values. The series follows an autoregressive model of order p, say AR(p), if the value of the series at time $t > p$ is a linear function of the values observed in the previous *p* steps. We can describe the model in a matrix form as

$$x_j = X_j \beta_j + \varepsilon_j \tag{3}$$

where x_j is the vector $(x_{j(p+1)}, \ldots, x_{jn})^T$, X_j is the (n−p)×(p+1) regression matrix whose *t*th row is $(1, x_{j(t-1)}, \ldots, x_{j(t-p)})$ for $t > p$, β_j is the vector of autoregressive coefficients and ε_j is the vector of uncorrelated errors that are assumed normally distributed with expected value $E(\varepsilon_{jt}) = 0$ and variance $V(\varepsilon_{jt}) = \sigma_j^2$, for any time point t. Given the data, the model parameters can be estimated using standard Bayesian procedures, and details are described in [12].

To select the set of clusters, BCD uses a novel model-based Bayesian clustering procedure. A set of clusters $C_1, . , C_k., . ,C_c$, each consisting of m_k time series, is represented as a model M_C.

The time series assigned to each cluster are treated as independent realizations of the dynamic process represented by the cluster, which is described by an autoregressive equation. Each cluster C_k can be jointly modeled as

$$x_k = X_k \beta_k + \varepsilon_k$$

where the vector x_k and the matrix X_k are defined by stacking the m_k vectors x_{kj} and regression matrices X_{kj}, one for each time series, as follows

$$x_k = \begin{pmatrix} x_{k1} \\ \vdots \\ x_{km_k} \end{pmatrix} \qquad X_k = \begin{pmatrix} X_{k1} \\ \vdots \\ X_{km_k} \end{pmatrix}$$

Given a set of possible clustering models, the task is to rank them according to their posterior probability. The posterior probability of the model M_C is computed by Bayes Theorem as

$$P(M_C \mid x) \propto P(M_C) f(x \mid M_C) \tag{4}$$

where $P(M_C)$ is the prior probability of M_C and $f(x| M_C)$ is the marginal likelihood. Assuming independent uniform prior distributions on the model parameters and a symmetric Dirichlet distribution on the cluster probability p_k, the marginal likelihood of each cluster model M_C can be easily computed in a closed form by solving the integral:

$$f(x \mid M_C) = \int f(x \mid \theta_C) f(\theta_C) d\theta_C \tag{5}$$

where θ_C is the vector of parameters that describe the likelihood function, conditional on a clustering model M_C, and $f(\theta_C)$ is the prior density. In this way, each clustering model has an explicit probabilistic score and the model with maximum score can be found. In particular, $f(x| M_C)$ can be computed as

$$f(x \mid M_c) = \frac{\Gamma(1)}{\Gamma(1+m)} \times \prod_{k=1}^{c} \frac{\Gamma(m_k / m + m_k)}{\Gamma(m_k / m)} \frac{(\frac{RSS_k}{2})^{(q-n_k)/2} \Gamma(\frac{n_k - q}{2})}{(2\pi)^{(q-n_k)/2} \det(X_k^T X_k)^{(1/2)}} \tag{6}$$

where n_k is the dimension of the vector x_k, and $RSS_k = x_k^T (I_n - X_k (X_k^T X_k)^{-1} X_k^T) x_k$ is the residual sum of squares in cluster C_k. When all clustering models are *a priori* equally likely, the posterior probability $P(M_C| \ x)$ is proportional to the marginal likelihood $f(x|M_C)$, which becomes our probabilistic scoring metric.

As the number of clusters or subsets grows exponentially with the number of time series, BCD uses an agglomerative search strategy, which iteratively merges time series into clusters. The procedure starts by assuming that each of the m electricity load time series is generated by a different process. Thus, the initial model M_m consists of m clusters, one for each time series, with score $f(x|M_m)$. The next step is the computation of the marginal likelihood of the $m(m-1)$ models in which two of the m profiles are merged into one cluster. The model M_{m-1} with maximal marginal likelihood is chosen and the merging is rejected if $f(x \mid M_m) \geq f(x \mid M_{m-1})$ and the procedure stops. If $f(x \mid M_m) < f(x \mid M_{m-1})$, the merging is accepted and a cluster C_k merging the two time series is created. In such a way, the procedure is repeated on the new set of $m-1$ time series that consist of the remaining $m-2$ time series and the cluster profile.

Although the agglomerative strategy makes the search process feasible, the computational effort can be extremely demanding when the number of time series is large. To further reduce this effort, we use a heuristic strategy based on a measure of similarity between time series. The intuition behind this strategy is that the merging of two similar time series has better chances of increasing the marginal likelihood. The heuristic search starts by computing the $m(m-1)$ pair-wise similarity measures of the time series and selects the model M_{m-1} in which the two closest time series are merged into one cluster. If $f(x \mid M_m) < f(x \mid M_{m-1})$, the two time series are merged into a single cluster, a profile of this cluster is computed by averaging the two observed time series, and the procedure is repeated on the new set of $m-1$ time series. If this merging is rejected, the procedure is repeated on the two time series with the second highest similarity until an acceptable merging is found. If no acceptable merging is found, the procedure stops. Note that the clustering procedure is actually performed on the posterior probability of the model and the similarity measure is only used to increase the speed of the search process and to limit the risk of falling into local maxima.

Similarity measures of two time series implemented in BCD include Euclidean distance, correlation and Kullback–Leiber distance. In the numerical experiments, we have tried different distances and finally adopted the Euclidean distance of two time series.

3.3 Learning Algorithm: The SVR Network

SVR (or SVM) is a new and powerful machine learning technique for data classification and regression based on recent advances in statistical learning theory [14], [15]. Supposing that we are given training data $(x_1,y_1),\ldots(x_i,y_i),\ldots(x_n,y_n)$ where x_i are input patterns and y_i are the associated output value of x_i, the support vector regression solves an optimization problem

$$\min_{\omega,b,\xi,\xi^*} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

$$\text{Subject to } y_i - (\omega^T\phi(x_i) + b) \le \varepsilon + \xi_i^*, \qquad (7)$$

$$(\omega^T\phi(x_i) + b) - y_i \le \varepsilon + \xi_i,$$

$$\xi_i, \xi_i^* \ge 0,\ i=1,\ldots,n$$

where x_i is mapped to a higher dimensional space by the function Φ, and ξ_i* is slack variables of the upper training error (ξ_i is the lower) subject to the ε-insensitive tube $(\omega^T\phi(x_i) + b) - y_i \le \varepsilon$. The constant C>0 determines the trade off between the flatness and losses. The parameters which control regression quality are the cost of error C, the width of the tube ε, and the mapping function Φ.

The constraints of (7) imply that we put most data x_i in the tube ε. If x_i is not in the tube, there is an error ξ_i or ξ_i* which we tend to minimize in the objective function. SVR avoids under-fitting and over-fitting of the training data by minimizing the training error $C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$ as well as the regularization term $\omega^T\omega/2$. For traditional least-square regression, ε is always zero and data are not mapped into higher dimensional spaces. Hence, SVR is a more general and flexible treatment on regression problems.

Since Φ might map x_i to a high or infinite dimensional space, instead of solving ω for (7) in a high dimension, we deal with its dual problem, which can be derived using the Lagrange theory.

$$\max_{\alpha_i,\alpha_i^*} -\frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i - \alpha_i^*)^T Q(\alpha_j - \alpha_j^*) - \varepsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)$$

$$\text{Subject to, } \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0 \qquad (8)$$

$$0 \le \alpha_i, \alpha_i^* \le C,\ i=1,\ldots,n$$

where $Q_{ij} = \phi(x_i)^T\phi(x_j)$, α_i and α_i* are the Lagrange multipliers. However, this inner product may be expensive to compute because $\phi(x)$ has too many elements. Hence, we apply "kernel trick" to do the mapping implicitly. That is, to employ some special forms, inner products in a higher space yet can be calculated in the original space. Typical examples for the kernel functions are polynomial kernel $\phi(x_i)^T\phi(x_j) = (\gamma x_1^T x_2 + c_0)^d$ and RBF kernel $\phi(x_i)^T\phi(x_j) = e^{-\gamma(x_1-x_2)^2}$. They are inner products in a very high dimensional space (or infinite dimensional space) but can be computed efficiently by the kernel trick even without knowing $\phi(x)$.

As each data subset classified from the BCD is considered to be approximately stationary, 24 SVRs are applied to respectively fit the hourly electricity load profile data by taking advantage of all past information and similar dynamic properties (e.g. piece-wise stationarity). The next-day electricity load forecasting is conducted by the trained network with an acceptable level of accuracy in a voting manner in the BCD and SVRs. For numerical experiments in this paper, we use the software LIBSVM [16], which is a library for support vector machines, including the efficient implementation of solving (8).

4 Numerical Experiments

4.1 Data Collection and Preprocess

The daily electricity load in New York City and weather data observed at Central Park have been considered for the study. Two testing sets have been selected to forecast and validate the performance of the proposed model. The first one corresponds to January and February and the second one corresponds to July and August. These months are all typical months with high demand. The hourly data used to forecast the two testing sets are from January 1, 2001 to December 31, 2003 and July 1, 2001 to June 30, 2004. The test sets are completely separate from the training sets and are not used during the learning procedure.

4.2 Numerical Results

The criteria to compare the performance are the mean absolute error (MAE) and mean absolute percentage error (MAPE) in this paper. For comparative study, we calculate the MAE and MAPE of the forecasting published by New York ISO in the same period. Moreover, a model using SVR network without the gating stage of BCD is also built and studied.

Numerical results with the proposed model are presented. For simplicity, we call the proposed model MLF. The hourly MAE and MAPE for the two test sets with the three different methods are shown in Tables 4, 5 and 6. From the three tables, clearly the proposed model outperforms all others in almost all the situations.

Figs. 3-4 show the forecasting and the actual load curves. To illustrate the variety of load more clearly, we respectively plot different parts of the load curves in the first testing set for presentation. It can be seen that the proposed model well predicts the trend of the price curve generally.

Table 4. Results for all days of each testing set

	First testing set		Second testing set	
	MAE	MAPE	MAE	MAPE
ISO	135.71	2.42	214.40	3.16
Single SVR	125.22	2.15	226.87	3.27
MLF	79.15	1.39	178.21	2.51

Table 5. Results for the normal days (work days)

	First testing set		Second testing set	
	MAE	MAPE	MAE	MAPE
ISO	119.93	2.04	208.93	2.94
Single SVR	120.49	1.96	215.66	2.91
MLF	69.79	1.16	170.57	2.31

Table 6. Results for anomalous days (including weekend and holidays)

	First testing set		Second testing set	
	MAE	MAPE	MAE	MAPE
ISO	165.02	3.12	226.77	3.65
Single SVR	134.01	2.50	252.24	4.08
MLF	96.54	1.80	195.49	2.97

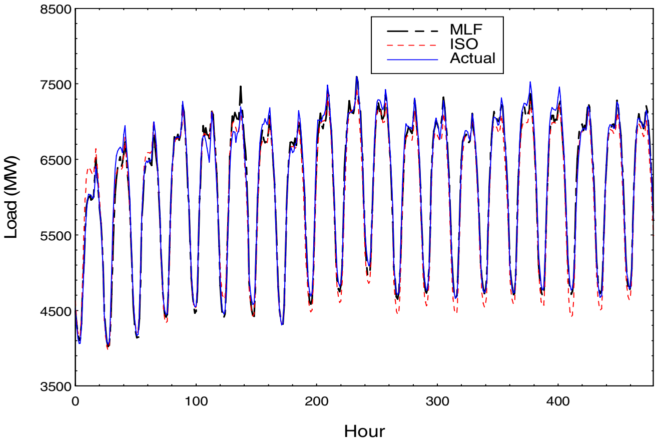


Fig. 3. Forecasting results for regular days in January 2004

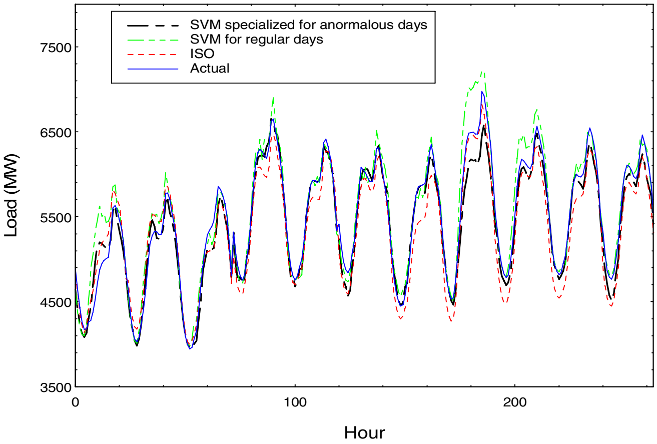


Fig. 4. Forecasting results for weekend and holidays in January 2004

5 Conclusion

In this paper, an integrated machine learning forecasting model, based on BCD and SVR has been developed to predict next day electricity load curve. The proposed method was applied to the prediction of the load curves in New York City, which demonstrates its effectiveness and efficiency of the learning and prediction in contrast to others.

Although many efforts have been make to the forecasting of electricity load in the past years. This problem is still such a difficult task that a comprehensive and general solution is far from easy. For a specific system, the best performances can be achieved only if a deep investigation of the inherent characteristics for the system has been carried out. In the future, we will further study to incorporate operational and market factors in our model to improve the prediction accuracy.

References

1. Henrique Steinherz Hippert, Carlos Eduardo Pedreira, and Reinaldo Castro So: Neural Networks for Short-Term Load Forecasting: A Review and Evaluation. IEEE Trans. Power Systems, vol. 16. (2001) 44-55
2. Haida T., Muto S.: Regression based peak load forecasting using a transformation technique. IEEE Trans. Power Systems, vol. 9. (1994) 1788-1794
3. Huang S. J., Shih K. R.: Short-term load forecasting via ARMA model identification including nongaussian process considerations. IEEE Trans. Power Systems, vol. 18. (2003) 673-679
4. Box G. E. P., Jenkins G. M.: Time series analysis – forecasting and control. Holden-day, San Francisco (1976)
5. Czernichow T., Piras A., Imhof K., Caire P., Jaccard Y., Dorizzi B., Germond A.: Short term electrical load forecasting with artificial neural networks. Engineering Intelligent Systems, vol. 2. (1996) 85-99

6. Fan S., Mao C. X., Chen L. N.: Peak load forecasting using the Self-Organizing Map. Advances in Neural Network-ISNN 2005, Springer-Verlag, Berlin Heidelberg New York, Part III, (2005) 640-647
7. Song K. B., Baek Y. S., Hong D. H., Jang G.: Short-term load forecasting for the holidays using fuzzy linear regression method. IEEE Trans. Power Systems, vol. 20. (2005) 96-101
8. Fidalgo J.N., Pecas Lopes J.A.: Load forecasting performance enhancement when facing anomalous events. IEEE Trans. Power Systems, vol. 20. (2005) 408-415
9. Chen B. -J., Chang M. -W., Lin C.-J.: Load forecasting using support vector machines: a study on EUNITE competition 2001. IEEE Trans. Power Systems, vol. 19. (2004) 1821-1830
10. Fan S., Chen L.: Short-Term Load Forecasting Based on an Adaptive Hybrid Method. IEEE Trans. Power Systems. Vol. 21. (2006) 392-401
11. Ramoni M., Sebastiani P., Cohen P.: Bayesian Clustering by Dynamics. Machine Learning. vol. 47. (2002) 91-121
12. Sebastiani P., Ramoni M.: Clustering continuous time series. Proc Eighteenth Intl Conf on Machine Learning (ICML-2001) 497-504
13. http://www.nyiso.com [Online]
14. Cortes C., Vapnik V.: Support-vector network. Machine Learning. vol. 20. (1995) 273-297
15. Cristianini N., Shawe-Tylor J.: An introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, Cambridge, United Kingdom, 2000
16. Chang C.-C., Lin C.-J.: (2001) LIBSVM: A library for Support Vector Machines. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

Short-Term Load Forecasting Using Multiscale BiLinear Recurrent Neural Network with an Adaptive Learning Algorithm

Chung Nguyen Tran, Dong-Chul Park, and Hwan-Soo Choi

ICRL, Dept. of Information Engineering, Myong Ji University, Korea
{tnchung, parkd, hschoi}mju.ac.kr

Abstract. In this paper, a short-term load forecasting model using a Multiscale BiLinear Recurrent Neural Network with an adaptive learning algorithm (M-BLRNN(AL)) is proposed. The proposed M-BLRNN(AL) model is based on a wavelet-based neural network architecture formulated by a combination of several individual BLRNN models. The wavelet transform adopted in the M-BLRNN(AL) is employed to decompose the load curve into a mutiresolution representation. Each individual BLRNN model is used to forecast the load signal at each resolution level obtained by the wavelet transform. The learning process is further improved by applying an adaptive learning algorithm at each resolution level. Experiments and results on load data from the North-American Electric Utility (NAEU) show that the proposed M-BLRNN(AL) model converges faster and archives better forecasting performance in comparison with other conventional models.

1 Introduction

Electric load forecasting has received considerable attention from many researchers in recent years. Forecasting of electricity load can be performed by approximating an unknown nonlinear function of load data and other exogenous variables such as weather variables. Traditionally, statistical models such as the autoregressive model [1], the linear regression model [2], and the autoregressive moving average (ARMA) [3] have been widely used in practice because of their simplicity. However, these statistical models are based on linear analysis techniques. As such, they may not be suitable for load forecasting since models based on a linear analysis for approximating a nonlinear function often lead to inaccurate forecasting.

In recent years, various nonlinear-based models have been proposed for load forecasting. Among these models, neural network (NN)-based models are the favored choice for load forecasting because of their universal approximation abilities. Neural networks have the capacity not only to model time series load curves but also to model an unspecified nonlinear relationship between a load series and weather variables[4,5,6,7]. Comprehensive reviews of the application of neural networks to load forecasting in most recent studies show that neural

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 964–973, 2006.

network-based models give usable results and have been well accepted in practice by many utilities [8]. However, due to the very high cost associated with errors in practice, the development of more efficient load forecasting models continues to attract much attention.

In this paper, a short-term load forecasting model based on a Multiscale BiLinear Recurrent Neural Network with an adaptive learning algorithm (M-BLRNN(AL)) is proposed. The proposed M-BLRNN(AL) model is based on a wavelet-based neural network architecture formulated by a combination of several individual BLRNN models [9]. Each individual BLRNN model is used to forecast the load signal at a certain level obtained by a wavelet transform [10]. By employing the wavelet transform to decompose a load curve into multiresolution representations, a complex load curve can be simplified by several simpler sub-curves at each resolution level. By doing so, difficult forecasting tasks associated with the original load curve can be simplified by forecasting decomposed sub-curves at each resolution level. The learning speed and forecasting performance are further improved by applying an adaptive learning algorithm at each resolution level. The adaptive learning algorithm adopted in the M-BLRNN(AL) employs an adjustable activation function at each resolution level. By iteratively adapting the shape of the activation function to the range of load curves at each resolution level, the learning process can learn the underlying relationship between the input and output at each resolution level more efficiently. Thus, the M-BLRNN(AL) converges faster and archives better forecasting performance in comparison with other conventional models.

The remainder of this paper is organized as follows: Section 2 presents a review of the multiresolution analysis with the wavelet transform. A brief review of the BLRNN is given in Section 3. The M-BLRNN(AL) model and its adaptive learning algorithm are presented in Section 4. Section 5 presents experiments and results on a short-term load forecasting problem using the M-BLRNN(AL) model, including a performance comparison with other conventional models. Conclusions and additional remarks are given in Section 6.

2 Multiresolution Wavelet Analysis

The wavelet transform [10], a novel technology developed in the signal processing community, has received much attention from neural network researchers in recent years. Several NN models based on a multiresolution analysis using a wavelet transform have been proposed for time series prediction [11] and signal filtering [12]. In order to conduct a time series analysis, use of the discrete wavelet transform has been proposed [13]. More recently, the so-called *à* trous wavelet transform has been proposed. This approach produces a "smooth" approximation by filling the "gap" caused by decimation, using redundant information from the original signal [14].

The calculation of the *à* trous wavelet transform can be described as follows: First, a low-pass filter is used to suppress the high frequency components of a signal while allowing the low frequency components to pass through. A scaling

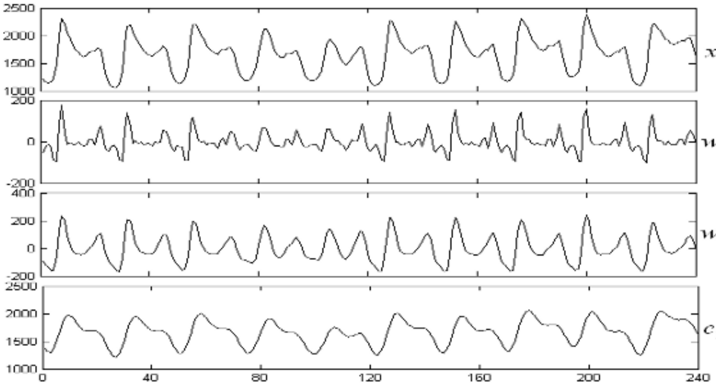


Fig. 1. Example of wavelet and scaling coefficients for a electric load data

function associated with the low-pass filter is then used to calculate the average of elements, which results in a smoother signal.

The smoothed data $c_j(t)$ at given resolution j can be obtained by performing successive convolutions with the discrete low-pass filter h,

$$c_j(t) = \sum_k h(k) c_{j-1}(t + 2^{j-1}k) \tag{1}$$

where h is a discrete low-pass filter associated with the scaling function and $c_0(t)$ is the original signal. In order to deal with time series forecasting, a nonsymmetric filter defined as $(\frac{1}{2}, \frac{1}{2})$ is employed for wavelet calculation.

From the sequence of the smoothing of the signal, the wavelet coefficients are obtained by calculating the difference between successive smoothed versions:

$$w_j(t) = c_{j-1}(t) - c_j(t) \tag{2}$$

By consequently expanding the original signal from the coarsest resolution level to the finest resolution level, the original signal can be expressed in terms of the wavelet coefficients and the scaling coefficients as follow:

$$c_0(t) = c_J(t) + \sum_{j=1}^{J} w_j(t) \tag{3}$$

where J is the number of resolutions and $c_J(t)$ is the finest version of the signal. Eq.(3) also provides a reconstruction formula for the original signal.

Fig. 1 shows an example of the wavelet coefficients and the scaling coefficients for two levels of resolution for the hourly electric load data of the North American Electric Utility (NAEU) from May 1, 1985 to May 10, 1985. From the top to the bottom are the original signal, two levels of the wavelet coefficients, and the finest scaling coefficients, respectively.

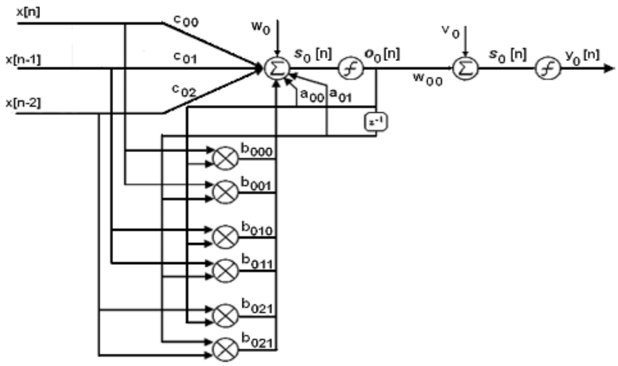


Fig. 2. Simple BLRNN with structure 3-1-1 and 2 recursion lines

3 BiLinear Recurrent Neural Networks

The BLRNN was first introduced by Park and Zhu [9]. It has been successfully applied in modeling time-series data [9,15]. Fig. 2 illustrates a simple 3-1-1 BLRNN with 2 feedback taps.

Assume that the input signal and the nonlinear integration of the input signal to hidden neurons are defined as:

$$\begin{aligned}\boldsymbol{X}[n] &= [x[n], x[n-1], ..., x[n-K]]^T \\ \boldsymbol{O}[n] &= [o_1[n], o_2[n], ..., o_M[n]]^T\end{aligned}$$

where T denotes the transpose of a vector or matrix and the recurrent term is a $M \times K$ matrix defined as:

$$\boldsymbol{Z}_p[n] = [o_p[n-1], o_p[n-2], ..., o_p[n-K]]$$

The output value of a bilinear recurrent neuron is computed by the following equation:

$$\begin{aligned}s_p[n] &= w_p + \sum_{k_1=0}^{N-1} a_{pk_1} o_p[n-k_1] \\ &+ \sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1} b_{pk_1k_2} o_p[n-k_1]x[n-k_2] \\ &+ \sum_{k_2=0}^{N-1} c_{pk_2} x[n-k_2] \\ &= w_p + \boldsymbol{A}_p^T \boldsymbol{Z}_p^T[n] + \boldsymbol{Z}_p[n]\boldsymbol{B}_p^T\boldsymbol{X}[n] + \boldsymbol{C}_p^T\boldsymbol{X}[n]\end{aligned} \tag{4}$$

where w_p is the weight of the bias neuron. $\boldsymbol{A}_p$ is the weight vector for the recurrent portion, $\boldsymbol{B}_p$ is the weight matrix for the bilinear recurrent portion,

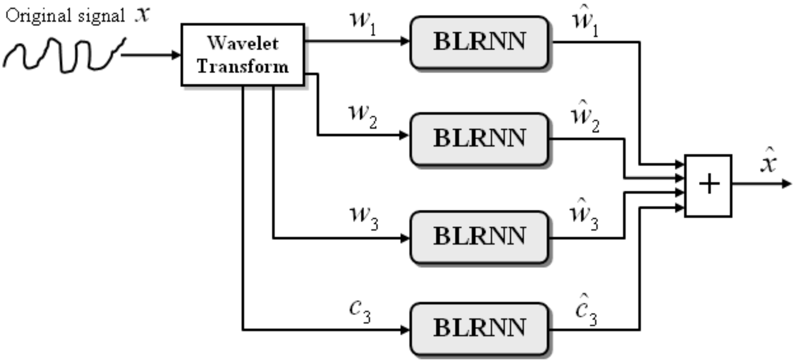


Fig. 3. Example of Multiscale BiLinear Recurrent Neural Network with 3 resolution levels

and $\boldsymbol{C}_p$ is the weight vector for the feedforward portion, and $p = 1, 2..., M$. Let ϕ be the activation function of the hidden neuron; the output of the p^{th} hidden neuron is then:

$$o_p[n] = \phi(s_p[n]) \tag{5}$$

From the hidden layer to the output layer, the output value is the same as that of a traditional feedforward-type neuron network:

$$s_l[n] = v_l + \sum_{p=0}^{N_h-1} w_{lp} o_p[n] \tag{6}$$

where v_l is the weight of the bias neuron, w_{lp} is the weight between the hidden and the output neurons, and N_h is the number of hidden neurons. The final output is obtained by applying the activation function

$$y_l[n] = \phi(s_l[n]) \tag{7}$$

More detailed information on the BLRNN and its learning algorithm can be found in [9,15].

4 Multiscale BiLinear Recurrent Neural Network with an Adaptive Learning Algorithm

The M-BLRNN(AL) is based on a wavelet-based neural network architecture formulated by a combination of several individual BLRNN models. Each BLRNN model is used to forecast a signal at each resolution level obtained by the wavelet transform. Fig. 3 illustrates an example of a M-BLRNN(AL) with three resolution levels.

As shown in Fig. 3, the load forecasting can be performed based on three separate stages. In the first stage, the original load is decomposed into the wavelet coefficients and the scaling coefficients. In the second stage, coefficients at each resolution level are forecasted by an individual BLRNN model. In the final stage, all of the forecasting results from individual BLRNNs are combined using the reconstruction formula of Eq.(3):

$$\hat{x}(t) = \hat{c}_J(t) + \sum_{j=1}^{J} \hat{w}_j(t) \tag{8}$$

where $\hat{c}_J(t)$, $\hat{w}_j(t)$, and $\hat{x}(t)$ represent the predicted values of the finest scaling coefficients, the predicted values of the wavelet coefficients at level j, and the predicted values of the original time series signal, respectively.

To further improve the convergence speed and forecasting performance, an adaptive learning algorithm is employed for training the M-BLRNN(AL) model. The adaptive learning algorithm adopted in the M-BLRNN(AL) uses an adjustable activation function at each resolution level. Typically, the activation function used in Eq.(5) and Eq.(7) is a logistic function defined as $\phi(x) = 1/(1+exp(-\lambda x))$. The slope parameter λ used in the adaptive learning algorithm is iteratively adapted at each training step using the gradient-descent method with respect to the characteristic of the input data.

Assume that the cost function is defined as

$$E = \frac{1}{2}\sum_{l}(t_l - y_l)^2 \tag{9}$$

At the output layer, the slope parameter λ_l at each output neuron l can be iteratively updated by

$$\begin{aligned} \lambda_l(n+1) &= \lambda_l(n) - \mu_\lambda \tfrac{\partial E}{\partial \lambda_l} \\ &= \lambda_l(n) + \mu_\lambda (t_l - y_l) \tfrac{s_l e^{-\lambda_l s_l}}{(1+e^{-\lambda_l s_l})^2} \end{aligned} \tag{10}$$

Similarly, at the hidden layer, the slope parameter λ_p at each hidden neuron p can be iteratively updated by

$$\begin{aligned} \lambda_p(n+1) &= \lambda_p(n) - \mu_\lambda \tfrac{\partial E}{\partial \lambda_p} \\ &= \lambda_p(n) + \left(\sum_l (t_l - y_l) \tfrac{\lambda_l e^{-\lambda_l s_l}}{(1+e^{-\lambda_l s_l})^2} w_{lp} \right) \tfrac{s_p e^{-\lambda_p s_p}}{(1+e^{-\lambda_p s_p})^2} \end{aligned} \tag{11}$$

The adaptation of the slope parameter adjusts the shape of the activation function to the range of the input and output values. As can be seen from Fig. 1, the characteristics of coefficients at each resolution level are different. Thus, by using the adaptive learning algorithm at each resolution level, the individual BLRNN model at each resolution level can learn the characteristics of the coefficients more efficiently. This implies that the proposed M-BLRNN(AL) model can yield better generalization performance and faster convergence than other conventional models.

Table 1. List of input variables for load forecasting models

Input	Variable name	Lagged value
1-5	Hourly load	1,2,3,24,168
6-10	Hourly temperature	1,2,3,24,168
11	Calendar variable	$\sin(2\pi t/24)$
12	Calendar variable	$\cos(2\pi t/24)$

5 Experiments and Results

The performance of the proposed M-BLRNN(AL) load forecasting model is evaluated and compared with other conventional models on the North-American Electric Utility (NAEU) data set. The NAEU data set consists of load and temperature data provided by the University of Washington at the following website:

http://www.ee.washington.edu/class/559/2002spr.

The temperature and load data were recorded at every hour of the day from January 1, 1985 to October 12, 1992, rendering 2,834 days of load and temperature data. Fig. 4 shows the hourly load demands from January 1, 1985 to December 31, 1985.

One of the most important steps in designing a neural network is choosing the input variables. The selection of appropriate input variables for a neural network can be performed based on an analysis of input data. As can be seen from Fig. 4, the load demand has multiple seasonal patterns such as daily and weekly periodicity: high demand in daytime and low demand at night time or high demand on weekdays and low demand on weekends. The load demand also has a strong correlation with the temperature. Low temperature results in high demand and high temperature results in low demand. Based on these seasonal patterns and the correlation analysis, the input variables for load forecasting models are selected as shown in Table 1.

The load forecasting for 1-24 hours ahead is performed using the recursive forecasting method. The forecasted output is fed back as the input for the next time-unit forecasting and all other network inputs are shifted back one time unit.

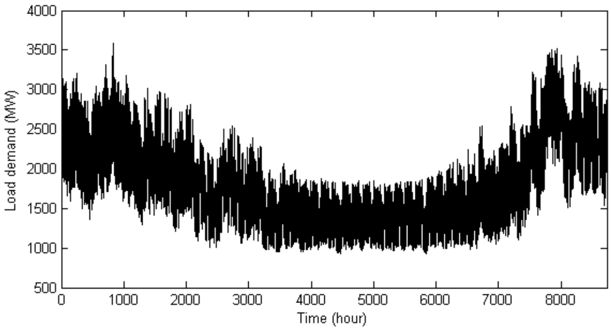


Fig. 4. Hourly load from January 1, 1985 to December 31, 1985

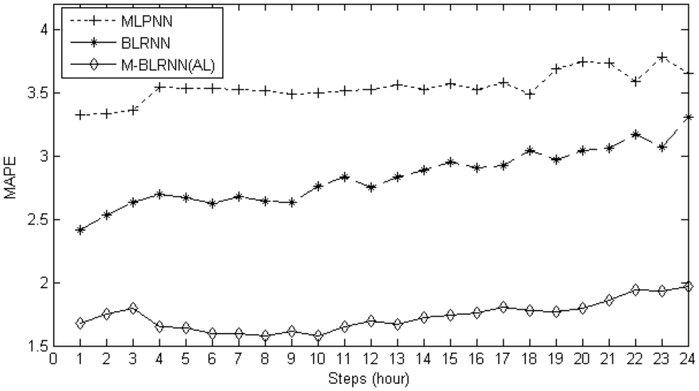
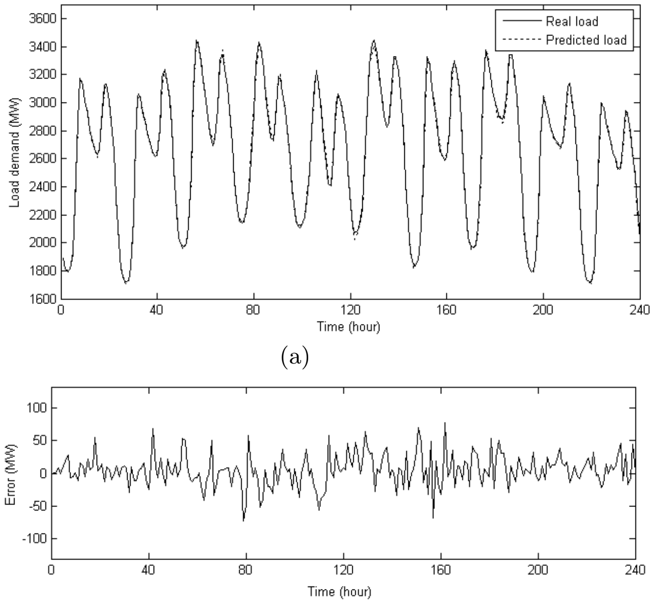


Fig. 5. 1-24 steps ahead of hourly forecasting performance in terms of MAPE

However, the future temperature is not available in practice when the recursive forecasting is performed. Therefore, it is necessary to estimate the temperature. In our experiments, the temperature was estimated from an average of the past temperature data.



(a)

(b)

Fig. 6. (a) Forecasting and real hourly load demand from January 15, 2002 to January 25, 2002 (b) Corresponding errors of forecasting results

The conventional MultiLayer Perceptron Type Neural Network (MLPNN) and the BLRNN were also employed in order to provide a comparison of the performance. All the above models utilized the input variables, as shown in Table 1. The MLPNN and the BLRNN were trained with 3,000 iterations, while the M-BLRNN(AL) was trained with 2,000 iterations. All the models were retrained at the end of each day to incorporate the most recent load information. All of the data used in our experiments were treated as normal working days. Holidays and anomalous days were not considered in this paper. The performance of the load forecasting models was evaluated in terms of the Mean Absolute Percentage Error (MAPE).

Fig. 5 shows the performance over 1-24 hours ahead for the short-term load forecasting using different models during the month of January 2002. As can been seen from Fig. 5, the proposed M-BLRNN(AL) model achieves significant improvement while requiring fewer iterations for training when compared with the conventional MLPNN and the BLRNN models. This implies that the M-BLRNN employing the wavelet-based neural architecture has robust capacity for load forecasting problems. Furthermore, by applying the adaptive learning algorithm, the M-BLRNN(AL) model converges faster than the MLPNN and the BLRNN models. This is an advantageous feature when addressing load forecasting problems in which load forecasting models are retrained regularly.

Figs. 6(a) shows the forecasting results at each hour from January 15, 2002 to January 25, 2002, which has a typical winter load profile. Fig. 6(b) plots the corresponding errors of forecasting results.

6 Conclusion

A short-term load forecasting model using a Multiscale BiLinear Recurrent Neural Network with an adaptive learning algorithm (M-BLRNN(AL)) is proposed in this paper. The proposed M-BLRNN(AL) model is formulated by a combination of several individual BLRNN models. The wavelet transform adopted in the proposed M-BLRNN(AL) is used to decompose the load profile into a multiresolution representation. Each individual BLRNN model is used to forecast the sub-profiles at each resolution level obtained by the wavelet transform. The convergence speed and forecasting performance of the M-BLRNN(AL) are further improved by applying an adaptive learning algorithm. When applied to load data from the NAEU, the proposed M-BLRNN(AL) model shows significant improvement in performance and faster convergence in comparison with the conventional MLPNN and the BLRNN. Thus, the M-BLRNN(AL) model can be used as an efficient tool for practical load forecasting problems.

Acknowledgment

This work was supported by the Korea Research Foundation Grant funded by the Korea Government (MOEHRD-KRF-2005-042-D00265).

References

1. Papalexopoulos, A.D., Hesterberg, T.C.: A Regression-Based Approach to Short-Term System Load Forecasting. IEEE Trans. on Power System 5(4) (1990) 1535-1547
2. Hyde, O., Hodnett, P.F.: An Adaptable Automated Procedure for Short-Term Electricity Load Forecasting. IEEE Trans. on Power System 12(1) (1997) 84-94
3. Huang, S.J., Shih, K.R.: Short-Term Load Forecasting via ARMA Model Identification Including Non-Gaussian Process Considerations. IEEE Trans. on Power System 18(2) (2003) 673-679
4. Park, D.C., El-Sharkawi, M.A., Marks II, R.J., Atlas, L.E., Damborg, M.J.: Electric Load Forecasting using An Artificial Neural Network. IEEE Trans. on Power System 6(2) (1991) 442-449
5. Park,D.C.,Park,T.,Choi,S.: Short-Term Electric Load Forecasting using Recurrent Neural Network. Proc. of ISAP'97, (1997) 367-371
6. Taylor, J.W., Buizza, R.: Neural Network Load Forecasting With Weather Ensemble Predictions. IEEE Trans. on Power System (2002) 17(3) 626-632
7. Mohan Saini, L., Kumar Soni, M.: Artificial Neural Network-Based Peak Load Forecasting using Conjugate Gradient Methods. IEEE Trans. on Power System 17(3) (2002) 907-912
8. Hippert, H.S., Pedreira, C.E., Souza, R.C.: Neural Networks for Short-Term Load Forecasting: A Review and Evaluation. IEEE Trans. on Power System 16(1) (2001) 44-55
9. Park, D.C., Zhu, Y.: Bilinear Recurrent Neural Network. IEEE ICNN, Vol. 3, (1994) 1459-1464
10. Mallat, S.G.: A Theory for Multiresolution Signal Decomposition: the Wavelet Representation. IEEE Trans. on Pattern Analysis and Machine Intelligence 11(7) (1989) 674-693
11. Liang, Y., Page, E.W.: Multiresolution Learning Paradigm and Signal Prediction. IEEE Trans. Sig. Proc. 45 (1997) 2858-2864.
12. Renaud, O., Starck, J.L., Murtagh, F.: Wavelet-Based Combined Signal Filtering and Prediction. IEEE Trans. on Systems, Man and Cybernetics 35(6) (2005) 1241-251
13. Shensa, M.J.: The Discrete Wavelet Transform: Wedding the À Trous and Mallat Algorithms. IEEE Trans. on Signal Processing 40(10) (1992) 2463-2482
14. Aussem, A., Murtagh, F.: A Neuro-wavelet Strategy for Web Traffic Forecasting. J. Offic. Statist., Vol. 1, (1998) 6587.
15. Park, D.C., Jeong, T.K.: Complex Bilinear Recurrent Neural Network for Equalization of a Satellite Channel. IEEE Trans on Neural Networks 13(3) (2002) 711-725

A New Approach to Load Forecasting: Using Semi-parametric Method and Neural Networks

Abhisek Ukil and Jaco Jordaan

Tshwane University of Technology
Nelson Mandela Drive, Pretoria, 0001, South Africa
{abhiukil, jakop_s2003}@yahoo.com

Abstract. A new approach to electrical load forecasting is investigated. The method is based on the semi-parametric spectral estimation method that is used to decompose a signal into a harmonic linear signal model and a non-linear part. A neural network is then used to predict the non-linear part. The final predicted signal is then found by adding the neural network predicted non-linear part and the linear part. The performance of the proposed method seems to be more robust than using only the raw load data.

1 Introduction

Load forecasting is used to estimate the electrical power demand. In the last few years, several techniques for short- and long- term load forecasting have been discussed, such as Kalman filters, regression algorithms and artificial neural networks [1]. A neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths and the processing performed at computing elements or nodes. Artificial neural networks generally consist of three layers: input, hidden and output. Each layer consists of one or more nodes. The inputs to each node in input and hidden layers are multiplied with proper weights and summed together. The weighted composite sum is passed through a proper transfer function whose output is the network output. Typical transfer functions are Sigmoid and Hyperbolic Tangent. For an example of a neural network, see Fig. 1.

The layout of the paper is as follows: in section 2 we introduce the proposed method of treating the load prediction problem, section 3 shows the numerical results obtained, and the paper ends with a conclusion.

2 Semi-parametric Method

When we want to fit a model to data from a power system, we many time have components in the data that are not directly part of the process we want to describe. If a model is fit to the data as it is, then the model parameters will be biased. We would have better estimates of the model parameters if we first remove the unwanted components (nuisance, bias, or non-linear components).

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 974–983, 2006.

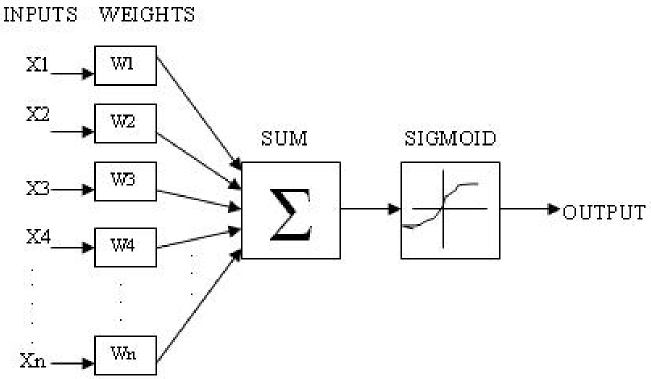


Fig. 1. Artificial Neural Network

This method has been used successfully in the field of spectral estimation in power systems when we analyse the measured signals on power transmission lines [2].

The new method we propose for load forecasting is based on a similar argument to separate the load data into linear and non-linear components. We name this method the Semi-Parametric method for harmonic content identification. We assume that there is an underlying linear part of the load data that could be represented with a sum of n damped exponential functions

$$y_L(k) = \sum_{i=1}^{n} A_i e^{j\theta_i} e^{(j2\pi f_i + d_i)Tk}, \tag{1}$$

where $y_L(k)$ is the $k-th$ sample of the linear part of the load signal, A_i is the amplitude, θ_i is the phase angle, f_i is the frequency, d_i is the damping and T is the sampling period. Since we work only with real signals, the complex exponential functions come in complex conjugate pairs. The equivalent Auto Regressive (AR) model of (1) is given by

$$y_L\left(k\right) = -\sum_{i=1}^{n} x_i y_L\left(k-i\right), \quad k = n+1 \ldots N, \tag{2}$$

with model parameters x_i, model order n, and $N = n+m$ number of samples in the data set. The model parameters x_i and model order n has to be estimated from the data.

We propose the following model to separate the linear and non-linear parts [2,3]:

$$y_L\left(k\right) = y\left(k\right) + \Delta y\left(k\right), \tag{3}$$

where $y\left(k\right)$ is the measured signal sample, $\Delta y\left(k\right) = E\left[\Delta y\left(k\right)\right] + \epsilon\left(k\right)$ is the residual component consisting of a non-zero time varying mean $E\left[\Delta y\left(k\right)\right]$ (nuisance or bias component) and noise $\epsilon\left(k\right)$. The mean of the residual component is represented by a Local Polynomial Approximation (LPA) model [4]. y_L is then the required linear signal that can be represented with a sum of damped exponentials (1). The LPA model is a moving window approach where a number of samples in the window are used to approximate (filter) one of the samples in the window (usually the first, last or middle sample). The LPA filtering of data was made popular by Savitsky and Golay [5,6].

By substituting eq. (3) in (2) we obtain

$$y\left(k\right) + \Delta y\left(k\right) = -\sum_{i=1}^{n} x_i \left[y\left(k-i\right) + \Delta y\left(k-i\right)\right]. \tag{4}$$

For $n+m$ samples we have:

$$\begin{bmatrix} y\left(n+1\right) + \Delta y\left(n+1\right) \\ y\left(n+2\right) + \Delta y\left(n+2\right) \\ \vdots \\ y\left(n+m\right) + \Delta y\left(n+m\right) \end{bmatrix} = -\begin{bmatrix} y\left(n\right) + \Delta y\left(n\right) & \cdots \\ y\left(n+1\right) + \Delta y\left(n+1\right) & \cdots \\ \vdots & \cdots \\ y\left(n+m-1\right) + \Delta y\left(n+m-1\right) & \cdots \end{bmatrix}$$

$$\begin{matrix} y\left(n-1\right) + \Delta y\left(n-1\right) \\ y\left(n\right) + \Delta y\left(n\right) \\ \vdots \\ y\left(n+m-2\right) + \Delta y\left(n+m-2\right) \end{matrix}$$

$$\begin{bmatrix} \cdots & y\left(1\right) + \Delta y\left(1\right) \\ \cdots & y\left(2\right) + \Delta y\left(2\right) \\ \ddots & \vdots \\ \cdots & y\left(m\right) + \Delta y\left(m\right) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \tag{5}$$

In matrix form the model is

$$\mathbf{b} + \Delta\mathbf{b} = -\mathbf{A}\mathbf{x} - \Delta\mathbf{A}\mathbf{x}, \tag{6}$$

where

$$\mathbf{b} = \begin{bmatrix} y\left(n+1\right) \\ y\left(n+2\right) \\ \vdots \\ y\left(n+m\right) \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} y\left(n\right) & y\left(n-1\right) & \cdots & y\left(1\right) \\ y\left(n+1\right) & y\left(n\right) & \cdots & y\left(2\right) \\ \vdots & \vdots & \ddots & \vdots \\ y\left(n+m-1\right) & y\left(n+m-2\right) & \cdots & y\left(m\right) \end{bmatrix}, \tag{7}$$

$$\Delta\mathbf{b} = \begin{bmatrix} \Delta y\left(n+1\right) \\ \Delta y\left(n+2\right) \\ \vdots \\ \Delta y\left(n+m\right) \end{bmatrix}, \Delta\mathbf{A} = \begin{bmatrix} \Delta y\left(n\right) & \Delta y\left(n-1\right) & \cdots & \Delta y\left(1\right) \\ \Delta y\left(n+1\right) & \Delta y\left(n\right) & \cdots & \Delta y\left(2\right) \\ \vdots & \vdots & \ddots & \vdots \\ \Delta y\left(n+m-1\right) & \Delta y\left(n+m-2\right) & \cdots & \Delta y\left(m\right) \end{bmatrix}. \tag{8}$$

The matrix signal model (6) can be rewritten in a different form and represented as

$$\mathbf{A}\mathbf{x} + \mathbf{b} + \left[\Delta\mathbf{b}\,\Delta\mathbf{A}\right] \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} = \mathbf{0}, \tag{9}$$

or

$$\mathbf{A}\mathbf{x} + \mathbf{b} + \mathbf{D}\left(\mathbf{x}\right) \varDelta\mathbf{y} = \mathbf{0}, \tag{10}$$

where the following transformation has been used:

$$\left[\varDelta\mathbf{b}\, \varDelta\mathbf{A}\right] \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} = \mathbf{D}\left(\mathbf{x}\right) \varDelta\mathbf{y} \tag{11}$$

or

$$\left[\begin{bmatrix} \varDelta y\left(n+1\right) \\ \varDelta y\left(n+2\right) \\ \vdots \\ \varDelta y\left(n+m\right) \end{bmatrix} \begin{bmatrix} \varDelta y\left(n\right) & \varDelta y\left(n-1\right) & \cdots & \varDelta y\left(1\right) \\ \varDelta y\left(n+1\right) & \varDelta y\left(n\right) & \cdots & \varDelta y\left(2\right) \\ \vdots & \vdots & \ddots & \vdots \\ \varDelta y\left(n+m-1\right) & \varDelta y\left(n+m-2\right) & \cdots & \varDelta y\left(m\right) \end{bmatrix} \right] \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} =$$

$$\begin{bmatrix} x_n & \cdots & x_1 & 1 & 0 & \cdots & 0 \\ 0 & x_n & \cdots & x_1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & x_n & \cdots & x_1 & 1 \end{bmatrix} \begin{bmatrix} \varDelta y\left(1\right) \\ \varDelta y\left(2\right) \\ \vdots \\ \varDelta y\left(n+m\right) \end{bmatrix}. \tag{12}$$

If the number of parameters in vector $\mathbf{x}$, (model order n) is not known in advance, the removal of the nuisance component and noise from the signal $y\left(k\right)$ is equivalent to estimating the residual $\varDelta y\left(k\right)$ and the model order n while fulfilling constraints (10). To solve the semi-parametric model, the second norm of the noise, plus a penalty term which puts a limit on the size of vector $\mathbf{x}$ is minimised. The following optimisation problem should be solved:

$$\begin{aligned} \underset{\mathbf{x},\varDelta\mathbf{y}}{min} \left\{ \frac{1}{2} \left\|\epsilon\right\|_2^2 + \frac{\mu}{2}\mathbf{x}^T\mathbf{x} \right\} &= \underset{\mathbf{x},\varDelta\mathbf{y}}{min} \left\{ \frac{1}{2} \left(\varDelta\mathbf{y} - E\left[\varDelta\mathbf{y}\right]\right)^T \left(\varDelta\mathbf{y} - E\left[\varDelta\mathbf{y}\right]\right) + \frac{\mu}{2}\mathbf{x}^T\mathbf{x} \right\} \\ &= \underset{\mathbf{x},\varDelta\mathbf{y}}{min} \left\{ \frac{1}{2}\varDelta\mathbf{y}^T\mathbf{W}\varDelta\mathbf{y} + \frac{\mu}{2}\mathbf{x}^T\mathbf{x} \right\} \end{aligned} \tag{13}$$

subject to the equality constraints $\mathbf{A}\mathbf{x} + \mathbf{b} + \mathbf{D}\left(\mathbf{x}\right) \varDelta\mathbf{y} = \mathbf{0}$,

where

$$\mathbf{W} = \left(\mathbf{I} - \mathbf{S}\right)^T \left(\mathbf{I} - \mathbf{S}\right), \tag{14}$$

$\mathbf{I}$ is the identity matrix, $\mathbf{S}$ is the LPA smoothing matrix used to estimate $E\left[\varDelta y\left(k\right)\right]$ as $\mathbf{S}\varDelta\mathbf{y}$, and μ is the Ridge regression factor used to control the size of vector $\mathbf{x}$ [7,8].

2.1 Estimation of the Harmonic Components

The next step is then to calculate the parameters of the harmonic components in eq. (1). We do this as follows [9,10]:

1. The coefficients x_i are those of the polynomial

$$\underline{H}(z) = 1 + \sum_{i=1}^{n} x_i \underline{z}^{-i}, \tag{15}$$

where $\underline{z}$ is a complex number

$$\underline{z} = e^{(j2\pi f + d)T}. \tag{16}$$

By determining the n roots, $\underline{z}_i$, $i = 1, 2, \ldots, n$, of eq. (15), and using eq. (16) for $\underline{z}$, we can calculate the values of the n frequencies and dampings of the harmonic components. It should be noted that we are using complex harmonic exponentials to estimate the input signal's linear component. However, the signals we measure in practice are real signals of the form

$$y(k) = \sum_{i=1}^{n/2} 2A_i e^{d_i T k} cos\left(2\pi f_i T k + \theta_i\right), \tag{17}$$

where A_i, θ_i, f_i and d_i are the same as defined for the complex harmonics in eq. (1). Therefore if we expect to have $\frac{n}{2}$ components in our real signal, there will be n complex harmonic exponentials, and thus will the AR model order be n. The complex harmonic exponentials will then always come in $\frac{n}{2}$ complex conjugate pairs.

2. To determine the n amplitudes A_i and phase angles θ_i, we substitute the linear component $y(k) + \Delta y(k)$, and the estimated frequencies and dampings into eq. (1). We obtain an overdetermined system of linear equations of $N \times n$ that can be solved using the least squares method:

$$y(k) + \Delta y(k) = \sum_{i=1}^{n} A_i e^{j\theta_i} e^{(j2\pi f_i + d_i)Tk}, \; k = 1, 2, \ldots, N. \tag{18}$$

2.2 Non-linear Part

The non-linear part (plus the noise), which could represent trends or other non-linearities in the power system, is then given by

$$y_N(k) = y(k) - y_L(k), \tag{19}$$

where $y_N(k)$ is the $k-th$ non-linear signal sample and $y(k)$ is the measured load sample. This non-linear part is then used to train a neural network. After the training is complete, the neural network could be used to predict the non-linear part. The linear part is calculated from the signal model (1), which is then added to the non-linear part to obtain the final predicted load values.

3 Numerical Results

For this experiment we used three types of neural networks, namely linear, Backpropagation Multi-layer Perceptron, and a Generalized Regression network [11]. The last network is a kind of Radial Basis network which is often used for function approximation and pattern matching. MATLAB Neural Network toolbox [12] was used for implementation.

Before the neural network is trained with the load data, some pre-processing is done on the data. First the data is scaled by the median of the data. Therefore, after prediction, the signal must be descaled by multiplying it again with the median. Then the scaled data is separated into a linear and a non-linear part.

The test data, shown in Fig. 2, contained 29 days of load values taken from a town at one hour intervals. This gives a total number of 696 data samples. We removed the last 120 data samples from the training set. These samples would then be used as testing data. Each sample is also classified according to the hour of the day that it was taken, and according to which day. The hours of the day are from one to 24, and the days from one (Monday) to seven (Sunday).

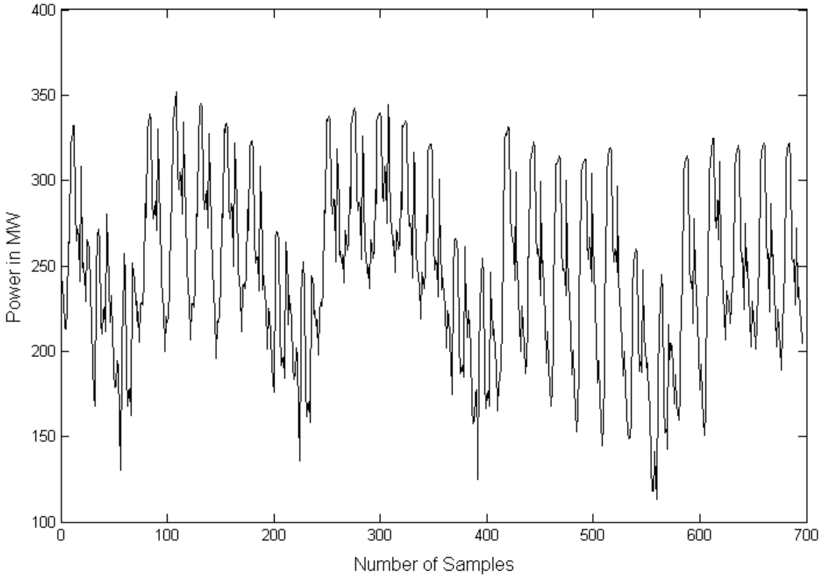


Fig. 2. Load of a Town

The data fed into the network could be constructed as follows: to predict the load of the next hour, load values of the previous hours are used. We can additionally also use the day and hour information. For example, this means that as inputs to the network, we could have k consecutive samples, and two additional input values representing the hour and day of the predicted $k+1-th$

sample. The network will then predict the output of the $k+1-th$ sample. We can also call the value of k : a delay of k samples.

To evaluate the performance of the different networks, we define a performance index, the Mean Absolute Prediction Error (MAPE):

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|t_i - p_i|}{t_i} \times 100\,, \tag{20}$$

where t_i is the $i-th$ sample of the true (measured) value of the load, p_i is the predicted load value of the network, and N is the total number of predicted samples. For this experiment, the last 24 hours of the 120 removed samples in the load set was used to test the different networks. Different values of delay was used, from five until 96.

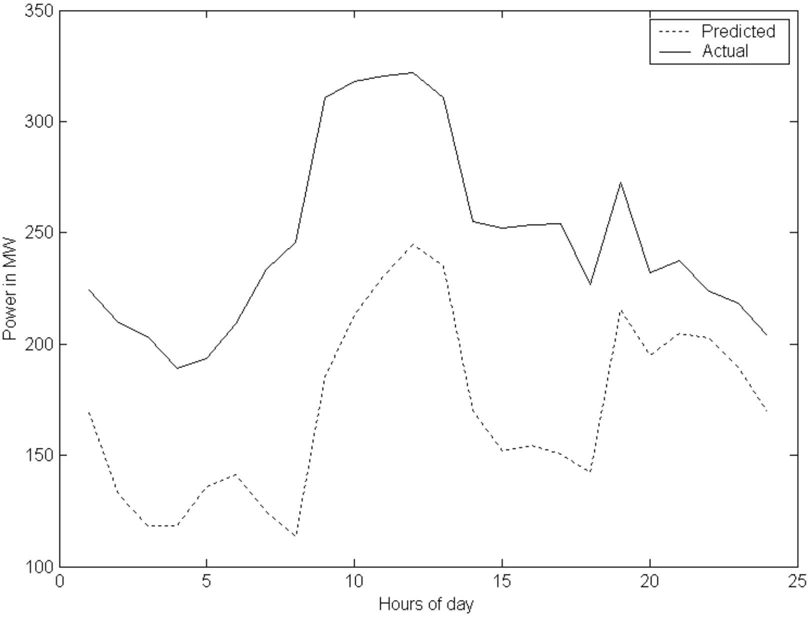


Fig. 3. Bad Performance of Method without Separating Data

We also tested the prediction method without splitting the data into linear and non-linear parts, and compared it to the proposed new method. The results of the performance index for each of the networks are shown in Tables 1, 2 and 3. It seems that the method without separating the data into different components performs slightly better than separating the data. In general the method with splitting the data performed well. There were a few occasions where the method without splitting the data had very bad performance, eg. Multi-Layer Perceptron

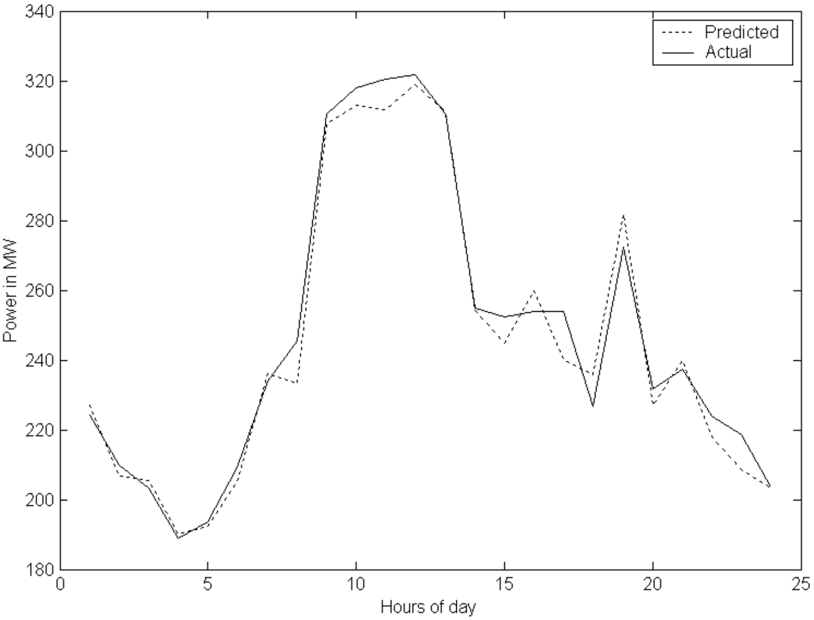


Fig. 4. Performance of Best Network

Table 1. MAPE for Linear Network

Separated into Linear / Non-Linear			Non-Separated	
Delay	Without day/hour	With day/hour	Without day/hour	With day/hour
5	11.1816	10.8449	6.5760	6.5667
10	9.9194	9.5566	5.9930	5.8607
16	9.0737	8.8913	6.4952	6.5134
96	2.0096	2.0749	2.0441	2.0046

Table 2. MAPE for Multi-layer Perceptron

Separated into Linear / Non-Linear			Non-Separated	
Delay	Without day/hour	With day/hour	Without day/hour	With day/hour
5	10.0207	6.3053	6.1400	6.7438
10	8.7914	5.6225	5.7735	5.1611
16	8.5937	6.6710	5.8434	30.5284

with delay 16, as can be seen in Fig. 3. The best network was without splitting the data, delay of 96. This is shown in Fig. 4. The best results for splitting the data is the linear network without day and hour information, delay of 96. This is shown in Fig. 5.

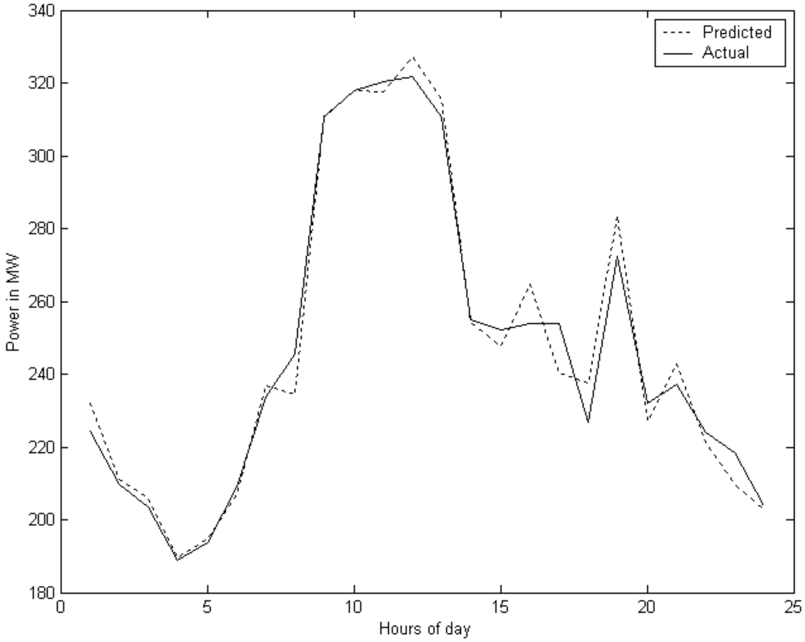


Fig. 5. Performance of Best Network for Splitting the Data

Table 3. MAPE for Generalized Regression Network

Separated into Linear / Non-Linear			Non-Separated	
Delay	Without day/hour	With day/hour	Without day/hour	With day/hour
5	13.3721	5.1269	11.1726	4.6823
10	13.2500	5.1501	11.4758	4.5819
16	13.0706	4.7016	11.8720	4.2966
96	8.3921	6.5287	7.6266	5.7553

4 Conclusion

The Semi-Parametric method for separating the electric load into a linear and non-linear part was introduced. A neural network was then used to do load forecasting based only on the non-linear part of the load. Afterwards the linear part was added to the predicted non-linear part of the neural network. We compared this method to the usual method without splitting the data. On average the method without splitting the data gave slightly better results, but there were occasions where this method produced very bad networks, whereas the newly introduced method generally performed well.

References

1. Bitzer, B., Rösser, F.: Intelligent Load Forecasting for Electrical Power System on Crete. In: UPEC 97 Universities Power Engineering Conference, UMIST-University of Manchester (1997)
2. Zivanovic, R.: Analysis of Recorded Transients on 765kV Lines with Shunt Reactors. In: Power Tech2005 Conference, St. Petersburg, Russia (2005)
3. Zivanovic, R., Schegner, P., Seifert, O., Pilz, G.: Identification of the Resonant-Grounded System Parameters by Evaluating Fault Measurement Records. IEEE Transactions on Power Delivery **19** (2004) 1085–1090
4. Jordaan, J., Zivanovic, R.: Time-varying Phasor Estimation in Power Systems by Using a Non-quadratic Criterium. Transactions of the South African Institute of Electrical Engineers (SAIEE) **95** (2004) 35–41 ERRATA: Vol. 94, No. 3, p.171-172, September 2004.
5. Gorry, P.: General Least-Squares Smoothing and Differentiation by the Convolution (Savitzky-Golay) Method. Analytical Chemistry **62** (1990) 570–573
6. Bialkowski, S.: Generalized Digital Smoothing Filters Made Easy by Matrix Calculations. Analytical Chemistry **61** (1989) 1308–1310
7. Draper, N., Smith, H.: Applied Regression Analysis. Second edn. John Wiley & Sons (1981)
8. Tibshirani, R.: Regression Shrinkage and Selection via the Lasso. Journal of the Royal Society. Series B (Methodological) **58** (1996) 267–288
9. Casar-Corredera, J., Alcásar-Fernándes, J., Hernándes-Gómez, L.: On 2-D Prony Methods. IEEE **CH2118-8/85/0000-0796 $1.00** (1985) 796–799
10. Zivanovic, R., Schegner, P.: Pre-filtering Improves Prony Analysis of Disturbance Records. In: Eighth International Conference on Developments in Power System Protection, Amsterdam, The Netherlands (2004)
11. Wasserman, P.: Advanced Methods in Neural Computing. Van Nostrand Reinhold, New York (1993)
12. Mathworks: MATLAB Documentation - Neural Network Toolbox. Version 6.5.0.180913a Release 13 edn. Mathworks Inc., Natick, MA (2002)

Research of Least Square Support Vector Machine Based on Chaotic Time Series in Power Load Forecasting Model

Wei Sun and Chenguang Yang

Department of Economy Management, North China Electric
Power University, Baoding 071003, Hebei, China
{bdsunwei, ycg1125}@126.com

Abstract. To predict short-term power load in an effective and fast way, the forecasting model of least square support vector machine (LSSVM) based on chaotic time series is established. According to A. Wolf method, Lyapunov exponents are worked out, and then the embedding dimension and time delay are also determined. And then the continuous power load data are transformed into data matrix by using the theory of phase-space reconstruction. Finally, LSSVM is used to train and predict the power load data. In order to prove the rationality of chosen dimension, another two random dimensions are selected to compare with the calculated dimension. And to prove the effectiveness and fast operating speed of the model, standard SVM algorithm and BP are used to compare with the model of LSSVM. The results show that the model is highly accurate and faster operating speed in short-term power load forecasting.

Keywords: Short-term power load, Lyapunov exponents, LSSVM, SVM, BP.

1 Introduction

The short-term power load forecasting is very significant to the electric network's reliable and economic running. With the development of electric market, people have been paying more and more attention to load prediction. How to make the prediction on the short-term power load exactly becomes a hot point [1].

Many scholars have studied more on the short-term power load forecasting, and they have raised many methods which can be classified into two sorts: one is a conventional method which takes advantage of time series, the other is the new artificial intelligence method which make use of Artificial Neural Network.

In the last twenty years, chaotic theory and statistic learning theory have become mature frequently and the usage of load prediction has become more and more mature. Least Square Support Vector Machines (LSSVM) is based on the statistic theory which is the improved algorithm of Support Vector Machines (SVM).

According to the characters of chaotic time series, the LSSVM prediction model based on chaotic time series is established. Then it is used in short-term power load forecasting of some electric networks to verify its effectiveness. As a result, the model with the embedding dimension, which is got through Lyapunov method, shows this model is more accurate than LSSVM with random embedding dimension and BP

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 984 – 993, 2006.

neural network, and the model also shows it expends less time than BP neural network and standard SVM.

2 Chaotic Time Series and Lyapunov Exponents

Base on Chaos theory, the drive factors have influenced each other in chaotic system. Therefore the digital points which are got according to time are relative. At present, people are employing the phase space delay coordinate reconstruction method in general to analyze the factors of serial dynamics. In fact, the phase space delay coordinate reconstruction method can expand the given time series $x_{1,}x_2,\cdots,x_{n-1},x_n,\cdots$ to three-dimensional and even higher dimensional space and the information which exposed sufficiently from time series can be classified and extracted [2], [3], [4], [5].

2.1 Reconstruction of Phase Space

In electric power system, actual loading series of single argument { $x(t_j)$ =1, 2, 3... n} can be got with the gap Δt. The structural character of system attractors is contained in this time series. The specific method, which can estimate the information of phase space reconstruction in single argument time series, is:

$$\begin{array}{ccccccc}
x(t_1) & x(t_2) & \cdots & x(t_j) & \cdots & x(t_n-(m-1)\tau) \\
x(t_1+\tau) & x(t_2+\tau) & \cdots & x(t_j+\tau) & \cdots & x(t_n-(m-2)\tau) \\
x(t_1+2\tau) & x(t_2+2\tau) & \cdots & x(t_j+2\tau) & \cdots & x(t_n-(m-3)\tau) \\
\cdots & \cdots & & \cdots & & \cdots \\
x(t_1+(m-1)) & x(t_2+(m-1)) & \cdots & x(t_j+(m-1)) & \cdots & x(t_n) \\
y(t_1) & y(t_2) & \cdots & y(t_j) & \cdots & y(t_n-(m-1)\tau)\,.
\end{array}$$

In this method, the time series can be continuatied to m-dimensional phase space. The time delay is $\tau = k\ \Delta t$ (k =1, 2....). In previous permutation, every column makes up a phase point of m-dimensional phase space. And each phase point has m components. These $n_p = n-(m-1)\tau$ phase points { $x(t_j)$, j =1, 2... n_p } make up a facies pattern in m-dimensional phase space. and the continuation of these phase points describes the evolutionary trace of system in the phase space.

2.2 Calculation of Lyapunov Exponents

A. Wolf submitted a method which is to extract maximal Lyapunov exponents in single argument time series. The process is:

(1) Reconstructing m-dimensional phase space with time series.
(2) Choosing minimal τ which marks the correlation among phase space.

(3) In the continuation *m*-dimensional phase space, the initial phase point $A(t_1)$ is chosen as a reference point. There are m components in the phase space, they are: $x(t_1), x(t_1+\tau), x(t_1+2\tau), \cdots x(t_1+(m-1))$. According to the following formula,

$$L_{nbt} = Min\left[\left\|Y_i - Y_j\right\|\right] \qquad i \neq j, \tag{1}$$

$B(t_1)$ which is the nearest neighborhood point to $A(t_1)$ can be impetrated. L_{nbt} which is assumed $L(t_1)$ means the distance between $A(t_1)$ and its nearest neighborhood point in Euclidean meaning. Suppose $t_2 = t_1 + k\Delta t$ with $k\Delta t$ as the step length and $A(t_1)$ evolves into $A(t_2)$, meanwhile $B(t_1)$ evolves into $B(t_2)$, then the distance $A(t_2)B(t_2) = l(t_2)$ is got. Let λ_1 represent the rate of exponential growth and $l(t_2) = L(t_1)2^{\lambda_2}$, then the following equation can be got.

$$\lambda_1 = \frac{1}{k(t_2 - t_1)} \log_2 (l(t_2) / L(t_1)) \qquad (\Delta t = 1), \tag{2}$$

λ is the Lyapunov exponent.

(4) Search a small neighborhood point $C(t_2)$ which subjects to the angle θ_1 in the nearest neighborhood points to $A(t_2)$ (If it can't meet the two conditions: small θ_1 and neighborhood, it should still choose $B(t_1)$). Suppose $t_3 = t_2 + k\Delta t$, $A(t_2)$ evolves into $A(t_3)$ and $C(t_2)$ evolves into $C(t_3)$ $A(t_2)C(t_2) = L(t_2)$ and $A(t_2)B(t_2) = l(t_2)$, then

$$\lambda_2 = \frac{1}{k} \log_2 (l(t_3) / L(t_2)). \tag{3}$$

It can not stop carrying out the previous steps until it reaches the end of point-group $\left\{X(t_j), j = 1, 2, \cdots, n_p\right\}$. Then choose the average of the calculated rates of exponential growth as the maximal estimated value of Lyapunov exponent. That is

$$LE_1(m) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{k} \log_2 \frac{l(t_i - 1)}{L(t_i - 1)} . \tag{4}$$

$N = n_p / k$ means total steps of step length.

(5) It can not stop increasing embedding dimension m in turn and carrying out the steps(3)-(4) until the estimated value $LE_1(m)$ of the exponent keeps stable and $LE_1(m_0) = LE_1(m_0+1) + LE_1(m_0+2) = \cdots = LE_1$. LE_1 is just the maximal Lyapunov exponent.

3 Least Square Support Vector Machine

Least square support vector machine is the improved algorithm of support vector machine [6], [7], [8], [9]. The following is the algorithm of linear case. Suppose l training samples $(x_1, y_1), \cdots (x_l, y_l) \in R^n \times R$.

Suppose the linear regression function is

$$f(x) = w^T x + b\ . \tag{5}$$

The structural risk function is introduced in and the regression problem is converted into the following quadratic programming.

$$\min \frac{1}{2}\|w\|^2 + \gamma \frac{1}{2}\sum_{i=1}^{l} \xi_i^2\ , \tag{6}$$

$$\text{s.t.}\quad y_i = w^T x_i + b + \xi\ , \qquad i = 1, \cdots, l\ . \tag{7}$$

Lagrange function is defined as

$$L = \frac{1}{2}\|w\|^2 + \gamma \frac{1}{2}\sum_{i=1}^{l} \xi_i^2 - \sum_{i=1}^{l} \alpha_i^2 (w^T x_i + b + \xi_i - y_i)\ . \tag{8}$$

According to KTT condition, there are

$$\frac{\partial L}{\partial w} = 0 \to w \sum_{i=1}^{l} \alpha_i x_i\ , \tag{9}$$

$$\frac{\partial L}{\partial b} = 0 \to w \sum_{i=1}^{l} \alpha_i = 0\ , \tag{10}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \to \alpha_i = \gamma \xi_i\ , \quad i = 1, \cdots, l\ , \tag{11}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \to w^T x_i + B + \xi_i - y_i = 0\ . \tag{12}$$

Function (5) to (8) can be described by the following linear equation,

$$\begin{bmatrix} I & 0 & 0 & -x \\ 0 & 0 & 0 & -\vec{1}^T \\ 0 & 0 & \gamma I & -I \\ x^T & \vec{1} & I & 0 \end{bmatrix} \begin{bmatrix} w \\ b \\ \xi \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ y \end{bmatrix}, \tag{13}$$

Here, $x=[x_1,\cdots,x_l]$, $y=[y_1,\cdots,y_l]$, $\vec{1}=[1,\cdots,1]$, $\xi=[\xi_1,\cdots,\xi_l]$, $\alpha=[\alpha_1,\cdots,\alpha_l]$. The final solution is

$$\begin{bmatrix} 0 & -\vec{1}^T \\ \vec{1} & x^T x+\gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} . \tag{14}$$

Here, $w=\sum_i \alpha_i x_i$,$\xi_i=\alpha_i/\gamma$.

As to the nonlinear regression problem, a nonlinear mapping ϕ is used to map the data to a high dimensional feature space. And then make linear regression in the feature space. The important problem is the choice of the kernel function $K(x,y)$, which makes $K(x_i,y_i)=\phi(x_i)^T\phi(x_j)$ [10], [11], [12]. The nonlinear regression is

$$f(x)=\sum_{i=1}^{l}\alpha_i K(x_i,x)+b \ . \tag{15}$$

4 LSSVM Based on Chaotic Time Series

The theories of time series and LSSVM are shown as follows.

4.1 LSSVM Based on Time Series

If time series $\{x_1,x_2\cdots x_N\}$ is given and the previous actual values of t time which are $x(1)$ 、 $x(2)$ $\cdots$ $x(t)$ are known. Then the forecasting value of $t+1$ time point can be got through the following map.

$$f:\boldsymbol{R}^m \to \boldsymbol{R} \tag{16}$$

Satisfy the following equation

$$x(t\hat{+}1)=f(x(t),x(t-1),\cdots x(t-(m-1))) . \tag{17}$$

$x(t\hat{+}1)$ is the predicted value of the $t+1$ time point, and m is the embedding dimension. Then we use LSSVM to make prediction.

4.2 Determination of Embedding Dimension

Lyapunov exponents attribute average speed of neighborhoods in system. A positive Lyapunov exponent is to attribute the segregation degree of average index number about two adjacent tracks while a negative Lyapunov exponent is to attribute the close degree. If a discrete nonlinear system is dissipative, relative stable and positive, Lyapunov exponent can be computed to judge whether the time series is chaotic or not [13], [14], [15]. The dimensional value is determined when Lyapunov exponents drive to be stationary. Then we can establish the model by combining embedding dimension with time series theory [16].

4.3 LSSVM Prediction Model

The given time series $\{x_1, x_2 \cdots x_N\}$ is separated into two parts. The previous n_{tr} data are used as training sample and the rest data as testing sample. The one-dimensional time series is converted into poly-dimensional matrix by reconstructing phase-space. Time delay is 1. The m-dimensional matrix is established as the following.

$$\boldsymbol{X} = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \\ x_2 & x_3 & \cdots & x_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n_{tr}-m} & x_{n_{tr}-m+1} & \cdots & x_{n_{tr}-1} \end{bmatrix}, \quad \boldsymbol{Y} = \begin{bmatrix} x_{m+1} \\ x_{m+2} \\ \vdots \\ x_{n_{tr}} \end{bmatrix}.$$

$\boldsymbol{X}$ is input matrix and $\boldsymbol{Y}$ is output matrix. $\boldsymbol{X}$ and $\boldsymbol{Y}$ satisfy (16). The predicted regression equation is

$$y_t = \sum_{i=1}^{n_{tr}-m} a_i K(x_i, x) + b \quad t = m+1, m+2 \cdots n_{t_r} . \tag{18}$$

The prediction model of the next time point is

$$y_{n_{tr}+1} = \sum_{i=1}^{n_{tr}-m} \alpha_i k(x_i, x_{n_{tr}-m+1}) + b . \tag{19}$$

and $X_{n_{tr}-m-1} = \left\{ x_{n_{tr}-m+1}, x_{n_{tr}-m+2}, \cdots x_{n_{tr}} \right\}$.

5 Application and Analysis

The whole process is shown as the following steps.

5.1 Samples Collection

Power load data in Hebei province is used to prove the effectiveness of the model. The power load data from 01:00 at 6/9/2005 to 24:00 at 7/18/2005 are as training sample and used to establish the single-variable time series $\{x(t_1), x(t_2), \cdots x(t_{960})\}$. And the power load data from 01:00 to 24:00 at 7/19/2005 as testing sample.

5.2 Chaos Analysis

For the training sample, τ =1 is chosen and Wolf method is used to compute Lyapunov exponents and embedding dimension. According to the theory 4.2, Lyapunov exponents λ begin to show stationary trend when the embedding dimension is 13. The power load time series shows chaotic character because λ >0. The embedding dimension is 13 and the number of phase points is 948. The above parameters are used to reconstruct the phase-space. The results are shown in Figure 1.

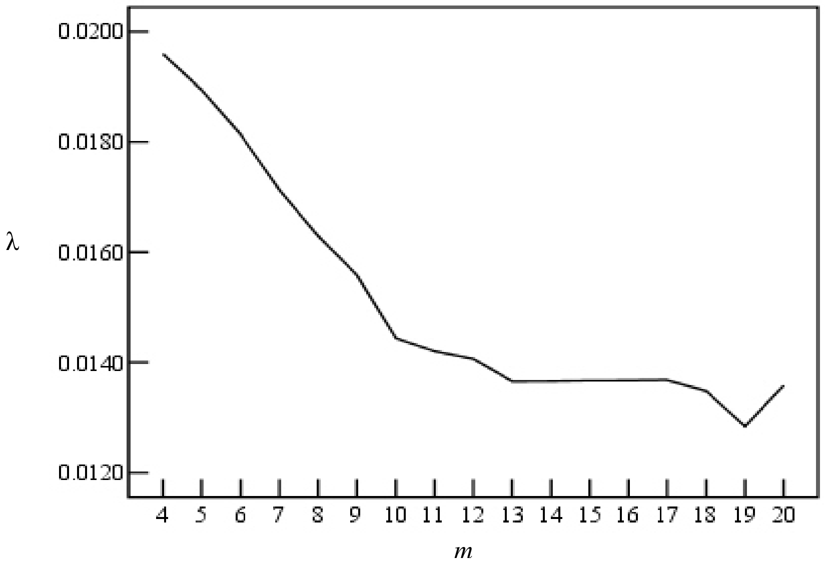


Fig. 1. λ (*Lyapunov exponent*) changes with m (*embedding dimension*). When embedding dimension is 13, Lyapunov exponents begin to show stable tendency.

5.3 Prediction Process

LSSVM is used after the samples are normalized. Matlab is used to compute the results. The computer with Pentium 4 1.7GHz CPU and 256MB inner memory is used in this experiment. Gauss kernel function is chosen as the kernel function [17].

$$K(x,x') = \exp(-\frac{\|x,x'\|^2}{2\sigma^2}) . \tag{20}$$

The parameters are chosen as the following: m =13, γ =1.2, σ =10. The other 12-dimensional matrix and 14-dimensional matrix are used as comparison.

BP algorithm is used to make prediction with sigmoid function. The network structure is 12-9-1. The system error is 0.001 and the maximal interactive time is 5000.

Standard SVM is also used to make prediction. Gauss kernel function is chosen as the kernel function. The parameters are chosen as the following: m =13, C=81.25, ε =0.045, σ^2 =2.23. The results are shown in Table 1.

5.4 Predicted Values and Evaluating Indicator

Relative error and root-mean-square relative error are used as evaluating indicators.

$$E_r = \frac{x_t - y_t}{x_t} \times 100\% \ , \ RMSRE = \sqrt{\frac{1}{N - n_{tr}} \sum_{t=n_{tr}}^{n} \left(\frac{x_t - y_t}{x_t} \right)^2} . \tag{21}$$

Table 1. Comparison of the predicted values and evaluating indicators

Time point	Original data	LSSVM (12) E_r	LSSVM (13) E_r	LSSVM (14) E_r	SVM (13) E_r	BP (13) E_r
01:00	417.29	2.97%	-2.54%	3.04%	-2.48%	3.68%
02:00	298.90	-2.69%	1.49%	2.25%	1.57%	2.97%
03:00	328.21	2.54%	-3.01%	4.57%	3.23%	-4.28%
04:00	328.21	3.09%	2.08%	-3.08%	2.00%	2.98%
05:00	398.53	2.67%	1.04%	1.79%	0.65%	2.58%
06:00	363.37	0.94%	-1.08%	2.39%	1.23%	-3.13%
07:00	345.79	-2.98%	0.98%	3.04%	0.46%	2.78%
08:00	310.62	3.69%	-1.78%	2.97%	-1.84%	2.52%
09:00	1057.29	5.61%	-3.67%	4.89%	2.49%	5.63%
10:00	1386.67	-2.79%	1.01%	2.54%	0.25%	-2.56%
11:00	1363.22	2.28%	2.12%	-3.44%	1.09%	0.21%
12:00	1439.41	-3.19%	4.79%	-6.85%	2.91%	-1.78%
13:00	940.07	-0.97%	-0.14%	1.78%	-0.98%	1.45%
14:00	904.91	3.04%	-2.36%	-2.26%	1.11%	6.58%
15:00	934.21	-2.17%	1.25%	-2.22%	1.36%	-3.25%
16:00	899.05	2.28%	-3.99%	4.02%	-2.24%	4.23%
17:00	963.52	-4.08%	2.12%	-3.02%	1.02%	3.48%
18:00	1339.78	2.29%	-1.24%	-2.88%	-0.21%	2.31%
19:00	1615.24	3.44%	2.34%	-3.65%	2.21%	3.67%
20:00	1727.77	2.30%	-4.59%	5.17%	-3.09%	-2.20%
21:00	1768.79	3.22%	0.11%	-3.03%	3.54%	6.87%
22:00	1398.39	4.56%	-2.23%	2.21%	-0.12%	2.28%
23:00	945.93	1.29%	-1.79%	3.22%	-1.21%	2.48%
24:00	287.18	3.82%	-2.75%	4.21%	-2.02%	-2.91%
RMSRE		0.0305	0.0243	0.0347	0.0191	0.0352

The results show:

(1) To see if the approach of embedding dimension chosen is reasonable or not, three cases are chosen as follows: 13-dimension, less than 13-dimension and larger than 13-dimension. If the results are measured by the standard which is less than or equal to 3%, the acceptable results in the condition of 13-dimension are 19 while 12-dimension are 14 and 14-dimension are 10. If the results are measured by root-mean-square relative error, RMSRE of 13-dimension is less than the other two dimensions. It can be seen from the above analysis that the predicting effectiveness of 13-dimension is better than other dimensions when LSSVM is used to make prediction.

(2) The comparison between LSSVM and BP is shown as the following. The relative error of predicted results by LSSVM has small rangeability. The maximal relative error is 4.79%and the value from the maximal relative error to the minimal relative error is 8.78%. On the contrary, the relative error of predicted results by BP has large rangeability. The maximal relative error is 6.87%and the value from the maximal relative error to the minimal relative error is 11.15%. If the results are measured by the standard which is less than or equal to 3%, the acceptable results of LSSVM are 19 while BP are 14. If the results are measured by root-mean-square relative error, RMSRE of LSSVM is less than BP. It can be seen from the above analysis that the predicting effectiveness of LSSVM is better than BP when the embedding dimension

is determined. As to operating time, LSSVM expends 98 seconds and BP expends 206 seconds. LSSVM needs less time than BP in computing so much data.

(3) The comparison between LSSVM and standard SVM is shown as the following. If the results are measured by the standard which is less than or equal to 3%, the acceptable results of LSSVM are 19 while standard SVM are 21. If the results are measured by root-mean-square relative error, RMSRE of LSSVM is larger than standard SVM. It shows that the predicting effectiveness of LSSVM is worse than standard SVM when the embedding dimension is determined. As to operating time, LSSVM expends 98 seconds and standard SVM expends 164 seconds. LSSVM needs less time than standard SVM in working out the data.

6 Conclusions

The results show that LSSVM based on chaotic time series has great effectiveness for short-term power load forecasting. And the conclusions are shown as the following:

(1) The power load data show apparent chaotic character. Chaotic time series is established and chaotic parameters are computed, then LSSVM prediction model is established to make prediction. The real load data prediction shows that the model is effective in short-term power load forecasting.

(2) The embedding dimension is chosen through Lyapunov method. The predicted results with chosen dimension and other random dimension are compared. The comparison shows that the approach is scientific and rational. The results show there is a suitable embedding dimension which is used to predict the power load effectively. The predicted values by the model with chosen dimension are highly accurate.

(3) In the condition of the same dimension, LSSVM is much higher than BP in accuracy. And LSSVM is less than standard SVM, but not so much missdistance in accuracy when the embedding dimension is determined.

(4) As to operating time, LSSVM is less than SVM and SVM is less than BP. Shorter operating time determines LSSVM is more fit for practice than SVM and BP.

Because the data of weather and temperature are hard to be acquired while the data of power load are easy to be acquired in fact and the shorter operating time, the model of LSSVM based on chaotic time series is more significant in the application than the models which need more power load data or the models which need the data of weather or temperature.

References

1. Niu, D.X., Cao, S.H., Zhao, Y.: Technology and Application of Power Load Forecasting.Beijing: China Power Press (1998)
2. Wen, Q., Zhang, Y.C., Chen, S.J.: The analysis Approach Based on Chaotic Time Series for Load Forecasting. Power System Technology, Vol.25 (2001) 13-16
3. Wolf, A., Swift, J. B., Swinney, H. L.: Determining Lyapunov Exponents from a time Series. Physics D., Vol.16 (1985)285-317
4. Lv, J.H., Lu, J.N., Chen, S.H.: Analysis and Application of Chaotic Time Series. Wuhan: Wuhan University Press (2002)

5. Liang, Z.S., Wang, L.M., Fu, D.P.: Short-Term Power Load Forecasting Based on Lyapunov Exponents. Proceeding of the CSEE, Vol.18 (1998)368-472
6. Vladimir, N.V., Zhang, X.G.: Nature of Statistics Theory. Beijing: Tsinghua University Press (2000)
7. Deng N.Y., Tian, Y.J.: the New Approach in Data Mining- Support Vector Machines. Science Press (2004)
8. Tan, D.N., Tan, D.H.: Small-Sample Machine Learning Theory-Statistical Learning Theory. Journal of Nanjing University of Science and Technology, Vol.25 (2001)108-112
9. Li, Y.C., Fang, T.J., Yu, E.K.: Study of Support Vector Machines for Short-Term Power Load Forecasting. Proceeding of the CSEE, Vol.25 (2003)55-59
10. Sun, D.S., Wu, J.P.: Application of LS-SVM to Prediction of Chaotic Time Series. Microcomputer Development, Vol.14 (2004) 21-23
11. Yan, W.W., Shao, H.H.: Application of Support Vector Machines and Least Square Support Vector Machines to Heart Disease Diagnoses. Control and Decision, Vol.3 (2003) 358-360
12. Zhu, J.Y., Chen, K.T., Zhang, H.X.: Study of Least Square Support Vector Machines. Computer Science, Vol. 30 (2003)157-159
13. Sun, K.H., Tan, G.Q., Sheng, L.Y.: Design and Implementation of Lyapunov Exponents Calculating Algorithm. Computer Engineering and Application, Vol.35 (2004)12-14
14. Li, G.H., Zhou, S.P., Xu, D.M.: Computing the Largest Lyapunov Exponents form Time Series. Journal of Applied Sciences, Vol.21 (2003)127-131
15. Li, T.Y., Liu, Z.F.: The Chaotic Property of Power Load and Its Forecasting. Proceeding of the CSEE, Vol.20 (2000)36-40
16. Zhang L., Liu, X.S., Yin, H.J.: Application of Support Vector Machines Based onTime Sequence in Power System Load Forecasting. Power System Technology, Vol.29 (2004) 38-41
17. Wang, X.D., Wang, J.P.: A Survey on Support Vector Machine Training and Testing Algorithms. Computer Engineering and Application, Vol.13 (2003)75-79

Solving Extended Linear Programming Problems Using a Class of Recurrent Neural Networks

Xiaolin Hu and Jun Wang

Department of Automation and Computer-Aided Engineering
The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong, China
{xlhu, jwang}@acae.cuhk.edu.hk

Abstract. Extended linear programming (ELP) is an extension of classic linear programming in which the decision vector varies within a set. In previous studies in the neural network community, such a set is often assumed to be a box set. In the paper, the ELP problem with a general polyhedral set is investigated, and three recurrent neural networks are proposed for solving the problem with different types of constraints classified by the presence of bound constraints and equality constraints. The neural networks are proved stable in the Lyapunov sense and globally convergent to the solution sets of corresponding ELP problems. Numerical simulations are provided to demonstrate the results.

1 Introduction

Extended linear programming (ELP) problems represent a class of linear programming problems in which the decision vector is not fixed, but varies in a set [1]. Such a situation may be encountered in many economic and social applications where the standard linear programming are employed. However, many effective methods for solving linear programming problems such as the simplex method and Karmarkar's method cannot be used to solve ELP problems. Researchers have to resort to other techniques. For example, by formulating the ELP problem, in which the decision vector of the conventional linear programming problem is allowed to vary within a box set, into a general linear variational inequality, He proposed an effective method to solve the problem [6].

In the past two decades, recurrent neural networks for solving optimization problems have attracted much attention. The theory, methodology, and applications of these neural networks have been widely investigated (e.g., see [2,3,4,5] and references therein). The impetus is two-fold. On one hand, the neural networks can be implemented in hardware and thus can solve problems in real-time. On the other hand, the dynamics of the neural networks may cast light to the development of new numerical algorithms. In 1997, Xia took an initiative to study ELP problems by using neural networks [7], focusing on solving a similar problem to that considered in [6]. After years, another recurrent neural network capable of solving the problems considered in [6] and [7] was developed by Gao in [8]. In this paper, we are concerned with solving the ELP problems using neural networks where the decision vector is allowed to vary within a general

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 994–1003, 2006.

polyhedral set rather than a box set. Several neural networks will be developed for this purpose with much effort in reducing the network complexity.

2 Problem Formulation and Preliminaries

Consider the following extended linear programming (ELP) problem:

$$\min_x\{\max_y y^T x\}, \quad \text{subject to } x \in X, y \in Y, Ax \in \Omega_1, By \in \Omega_2. \tag{1}$$

where $x, y \in R^n, A \in R^{h\times n}, B \in R^{r\times n}$, and X, Y, Ω_1, Ω_2 are box sets defined as

$$X = \{x \in R^n | \underline{x} \le x \le \overline{x}\},\ Y = \{y \in R^n | \underline{y} \le y \le \overline{y}\},$$
$$\Omega_1 = \{\xi \in R^h | \underline{\xi} \le \xi \le \overline{\xi}\},\ \Omega_2 = \{\eta \in R^r | \underline{\eta} \le \eta \le \overline{\eta}\}.$$

In above, $\underline{x}, \overline{x}, \underline{y}, \overline{y}, \underline{\xi}, \overline{\xi}, \underline{\eta}, \overline{\eta}$ are constants of appropriate dimensions. Without loss of generality, any component of $\underline{x}, \underline{\xi}, \underline{y}, \underline{\eta}$ can be $-\infty$, and any component of $\overline{x}, \overline{\xi}, \overline{y}, \overline{\eta}$ can be ∞. Clearly, when the decision vector y in (1) is a constant instead of a variable and $\underline{x} = 0, \overline{x} = \infty, \underline{\xi} = \overline{\xi} =$ contant, the ELP problem reduces to the classic linear programming problem. In (1), $x \in X, y \in Y$ are termed bound constraints and $Ax \in \Omega_1, By \in \Omega_2$ are termed general constraints in the paper. Though the bound constraints can be unified into general constraints, they are distinguished here because they can be handled with different techniques, which may lead to more efficient computational schemes. Throughout the paper, it is assumed that there always exists at least one solution (x^*, y^*) to the ELP problem.

In [7], a neural network for solving the ELP problem (1) with $X = R^n, B = I, Y = \Omega_2$ is proposed. The neural network is proved to be global convergent to its equilibrium points which correspond to the solutions of the ELP problems. Another ELP problem (1) with $B = I, Y = \Omega_2$ is formulated into a general linear variational inequality (GLVI) problem [6], which is solved by a neural network proposed later in [8]. In aforementioned studies [7, 6, 8], the decision vector y in the ELP problem is allowed to vary within a box set. In the paper, we consider the case in which the decision vector is allowed to vary within a general polyhedral set— this situation happens whenever $B \neq I$ in (1).

Before we move on to the next section, it is necessary to introduce the notion of GLVI and the corresponding neural network for solving it as they will play important roles in subsequent sections. Assume that $M, N \in R^{m\times n}$, $a, b \in R^m$ and $K \subset R^m$ is a nonempty closed convex set. The GLVI problem is to find $z^* \in R^n$ such that it satisfies $Nz^* + b \in K$ and

$$(Mz^* + a)^T(z - Nz^* - b) \ge 0 \quad \forall z \in K. \tag{2}$$

For solving the above GLVI problem, the following neural network is presented in [8]

$$\frac{dz}{dt} = \Lambda(N + M)^T(P_K((N - M)z + b - a) - Nz - b), \tag{3}$$

where Λ is a symmetric and positive definite matrix used to scaling the convergence rate of the neural network. The stability and convergence results, tailored from Theorems 3 and 4 in [8], are presented below.

Lemma 1. *The neural network in* (3) *is stable in the sense of Lyapunov and globally convergent to an exact solution of* (2) *when M^TN is positive semi-definite.*

The methodology of this study is to formulate the optimality conditions for ELP problems with different types of constraints into GLVI (2), and then use neural networks in the form of (3) (with different definitions of M, N, a, b, etc.) to solve the problems.

3 Neural Network Models for ELP

3.1 ELP Problems with General Constraints

Let's first consider the ELP problem (1) when all constraints are present. The following theorem establishes a necessary and sufficient optimality condition for problem (1) in this case.

Theorem 1. $(x^*, y^*) \in X \times Y$ *is a solution to problem* (1) *if and only if there exist* $s^* \in R^h, w^* \in R^r$ *such that* $Ax^* \in \Omega_1, By^* \in \Omega_2$, *and*

$$\begin{cases} (y^* - A^Ts^*)^T(x - x^*) \geq 0 \quad \forall x \in X, \\ (-x^* - B^Tw^*)^T(y - y^*) \geq 0 \quad \forall y \in Y, \\ (s^*)^T(s - Ax^*) \geq 0 \quad \forall s \in \Omega_1, \\ (w^*)^T(w - By^*) \geq 0 \quad \forall w \in \Omega_2. \end{cases} \tag{4}$$

Proof. In view of the fact that the general constraints in problem (1) can be expressed as

$$Ax - \alpha = 0, By - \beta = 0, \alpha \in \Omega_1, \beta \in \Omega_2,$$

define the Lagrangian function to problem (1) on $X \times Y \times \Omega_1 \times \Omega_2 \times R^h \times R^r$ as follows

$$L(x, y, \alpha, \beta, s, w) = y^Tx + s^T(\alpha - Ax) - w^T(\beta - By).$$

According to the well-known saddle point theorem, (x^*, y^*) is a solution to (1) if and only if there exist $\alpha^* \in \Omega_1, \beta^* \in \Omega_2, s^* \in R^h, w^* \in R^r$ such that

$$\begin{aligned} L(x^*, y, \alpha^*, \beta, s, w^*) \leq L(x^*, y^*, \alpha^*, \beta^*, s^*, w^*) \leq L(x, y^*, \alpha, \beta^*, s^*, w), \\ \forall x \in X, y \in Y, \alpha \in \Omega_1, \beta \in \Omega_2, s \in R^h, w \in R^r. \end{aligned} \tag{5}$$

The left inequality in (5) implies

$$\begin{aligned} &-(x^*)^Ty - s^T(\alpha^* - Ax^*) + (w^*)^T(\beta - By) \geq \\ &-(x^*)^Ty^* - (s^*)^T(\alpha^* - Ax^*) + (w^*)^T(\beta^* - By^*). \end{aligned}$$

Define a function

$$\phi(y, \beta, s) = -(x^*)^T y - s^T(\alpha^* - Ax^*) + (w^*)^T(\beta - By),$$

which is linear in y, s and β. Then

$$\phi(y^*, \beta^*, s^*) \leq \phi(y, \beta, s), \quad \forall y \in Y, \beta \in \Omega_2, s \in R^h.$$

According to [9], a necessary and sufficient condition for above inequality is as follows

$$\begin{cases} (-x^* - B^T w^*)^T(y - y^*) \geq 0, & \forall y \in Y, \\ (w^*)^T(\beta - \beta^*) \geq 0, & \forall \beta \in \Omega_2, \\ \alpha^* - Ax^* = 0. \end{cases} \tag{6}$$

Following a similar procedure as above we can derive the equivalent formulation of the right inequality in (1) as

$$\begin{cases} (y^* - A^T s^*)^T(x - x^*) \geq 0, & \forall x \in X, \\ (s^*)^T(\alpha - \alpha^*) \geq 0, & \forall \alpha \in \Omega_1, \\ \beta^* - By^* = 0. \end{cases} \tag{7}$$

By combining (6) and (7), and replacing α and β with notations s and w, respectively, we obtain the equivalent formulation of (5) as (4), which completes the proof.

Note that the optimality conditions in (4) can be expressed as: find z^* such that it satisfies $Nz^* \in K$ and

$$(Mz^* + a)^T(z - Nz^*) \geq 0 \quad \forall z \in K,$$

where

$$\begin{gathered} M = \begin{pmatrix} 0 & I & -A^T & 0 \\ -I & 0 & 0 & -B^T \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix}, a = \begin{pmatrix} p \\ q \\ 0 \\ 0 \end{pmatrix}, N = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ A & 0 & 0 & 0 \\ 0 & B & 0 & 0 \end{pmatrix}, \\ z = (x^T, y^T, s^T, w^T)^T, K = X \times Y \times \Omega_1 \times \Omega_2, \end{gathered} \tag{8}$$

with I denoting the identity matrices of proper dimensions. The above inequality is a special case of (2). Therefore, the following neural network, which is tailored from (3), can be used to solve the ELP problem (1) with parameters defined in (8)

$$\frac{dz}{dt} = \Lambda(N + M)^T(P_K((N - M)z - a) - Nz), \tag{9}$$

with Λ being a symmetric and positive definite matrix. It is easily verified that $M^T N$ is skew-symmetric and of course positive semi-definite. According to Lemma 1 we have the following stability results of the neural network.

Theorem 2. *The neural network in* (9) *with parameters defined in* (8) *is stable in the sense of Lyapunov and globally convergent to a solution of the ELP problem* (1).

3.2 ELP Problems Without Bound Constraints

Let's now consider the ELP problem (1) without any bound constraint, i.e.,

$$\min_x \{\max_y y^T x\}, \quad \text{subject to } Ax \in \Omega_1, By \in \Omega_2. \tag{10}$$

For solving this problem, the neural network in (9) with $X = R^n, Y = R^m$ can of course be utilized. The dimensionality of the neural network, defined as the dimensionality of the state of the neural network, is $2n+h+r$. However, we argue that a simplified neural network of (9) which is of $h + r$ dimensions (and as a result, would have lower hardware complexity) can be designed for this purpose. Let us first state the optimality conditions for the ELP problems in (10), which follows from Theorem 1 directly.

Corollary 1. *$(x^*, y^*) \in X \times Y$ is a solution to problem (10) if and only if there exist $s^* \in R^h, w^* \in R^r$ such that $Ax^* \in \Omega_1, By^* \in \Omega_2$, and*

$$\begin{cases} y^* - A^T s^* = 0, \\ -x^* - B^T w^* = 0, \\ (s^*)^T (s - Ax^*) \geq 0 \quad \forall s \in \Omega_1, \\ (w^*)^T (w - By^*) \geq 0 \quad \forall w \in \Omega_2. \end{cases} \tag{11}$$

From the first two equations in (11) we have

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} 0 & -B^T \\ A^T & 0 \end{pmatrix} \begin{pmatrix} s^* \\ w^* \end{pmatrix}.$$

Substituting it into the last two inequalities in (11) yields

$$\begin{pmatrix} s^* \\ w^* \end{pmatrix}^T \left[\begin{pmatrix} s \\ w \end{pmatrix} - \begin{pmatrix} 0 & -AB^T \\ BA^T & 0 \end{pmatrix} \begin{pmatrix} s^* \\ w^* \end{pmatrix} \right] \geq 0, \quad \forall s \in \Omega_1, w \in \Omega_2.$$

The above inequality is in the form of GLVI (2). According to Lemma 1, the following neural network can be used to solve the problem

$$\frac{dz}{dt} = \Lambda (N + I)^T (P_K((N - I)z) - Nz), \tag{12}$$

where

$$N = \begin{pmatrix} 0 & -AB^T \\ BA^T & 0 \end{pmatrix}, z = \begin{pmatrix} s \\ w \end{pmatrix}, K = \Omega_1 \times \Omega_2.$$

Since N is skew-symmetric, we have the following theorem.

Theorem 3. *The neural network in (12) is stable in the sense of Lyapunov and globally convergent to a solution of the ELP problem (10).*

One should notice that the state vector z of (12) consists of only s and w, but does not contain the variables of ELP, i.e., x and y. The relationship between the optimum to ELP (x^*, y^*) and the equilibrium point of (12) (s^*, w^*) is as follows: $x^* = -B^T w^*, y^* = A^T s^*$.

3.3 ELP Problems with Equality and Inequality Constraints and Without Bound Constraints

Consider another special case of (1) where bound constraints are absent but both equality and inequality constraints are present, i.e.,

$$\min_x\{\max_y y^T x\}, \quad \text{subject to } Ax \in \Omega_1, Cx = c, By \in \Omega_2, Dy = d. \tag{13}$$

where $C \in R^{p\times n}, c \in R^p, D \in R^{q\times n}, d \in R^q$ and the other parameters are the same as in (1). For a well defined problem, we should have $rank(C) = p, rank(D) = q$. It is noticed that the equality constraints in (13) can be converted to inequality constraints as $c \le Cx \le c$. Then a neural network in the form of (12) that is of $h+p+r+q$ dimensions can solve the problem elegantly. But this is not our concern in this subsection. Actually, what we will do next is to reduce the dimensions of the neural network in (12) (lower than $h+p+r+q$) for solving problem (13). Let us first present the optimality conditions for the problem.

Theorem 4. *$(x^*, y^*) \in X \times Y$ is a solution to problem* (13) *if and only if there exist $s^* \in R^h, w^* \in R^r, \lambda^* \in R^p, \mu^* \in R^q$ such that $Ax^* \in \Omega_1, By^* \in \Omega_2$, and*

$$\begin{cases} y^* - A^T s^* - C^T \lambda^* = 0, \\ -x^* - B^T w^* - D^T \mu^* = 0, \\ Cx^* = c, \\ Dy^* = d, \\ (s^*)^T (s - Ax^*) \ge 0 \quad \forall s \in \Omega_1, \\ (w^*)^T (w - By^*) \ge 0 \quad \forall w \in \Omega_2. \end{cases} \tag{14}$$

Proof. Define the Lagrangian function to problem (1) on $X \times Y \times \Omega_1 \times \Omega_2 \times R^h \times R^p \times R^r \times R^q$ as follows

$$L(x, y, \alpha, \beta, s, \lambda, w, \mu) = y^T x + s^T(\alpha - Ax) + \lambda^T(c - Cx) - w^T(\beta - By) - \mu^T(d - Dy).$$

According to the well-known saddle point theorem, (x^*, y^*) is a solution to (1) if and only if there exist $\alpha^* \in \Omega_1, \beta^* \in \Omega_2, s^* \in R^h, w^* \in R^r, \lambda^* \in R^p, \mu^* \in R^q$ such that

$$\begin{aligned} L(x^*, y, \alpha^*, \beta, s, \lambda, w^*, \mu^*) &\le L(x^*, y^*, \alpha^*, \beta^*, s^*, \lambda^*, w^*, \mu^*) \\ &\le L(x, y^*, \alpha, \beta^*, s^*, \lambda^*, w, \mu), \end{aligned}$$

$\forall x \in X, y \in Y, \alpha \in \Omega_1, \beta \in \Omega_2, s \in R^h, w \in R^r, \lambda \in R^p, \mu \in R^q$. The rest of the proof is similar to that of Theorem 1 and thus omitted.

Suppose that $p = q$ in problem (13). We propose the following neural network for solving (13)

$$\frac{dz}{dt} = \Lambda(N + I)^T (P_K((N - I)z + b) - Nz - b), \tag{15}$$

where Λ is symmetric and positive definite and

$$\begin{aligned} N &= \begin{pmatrix} 0 & -AB^T + AD^T(CD^T)^{-1}CB^T \\ BA^T - BC^T(DC^T)^{-1}DA^T & 0 \end{pmatrix}, \\ b &= \begin{pmatrix} AD^T(CD^T)^{-1}c \\ BC^T(DC^T)^{-1}d \end{pmatrix}, z = \begin{pmatrix} s \\ w \end{pmatrix}, K = \Omega_1 \times \Omega_2. \end{aligned} \tag{16}$$

Clearly, This neural network is of $h+r$ dimensions, lower than $h+r+p+q$. The properties of the neural network for solving (13) are revealed in Theorem 5.

Theorem 5. *If $p = q$ in (13), then the neural network in (15) is stable in the sense of Lyapunov and globally convergent to a solution of the ELP problem (13).*

Proof. The first two equations in (14) yields

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} 0 & -B^T \\ A^T & 0 \end{pmatrix} \begin{pmatrix} s^* \\ w^* \end{pmatrix} + \begin{pmatrix} 0 & -D^T \\ C^T & 0 \end{pmatrix} \begin{pmatrix} \lambda^* \\ \mu^* \end{pmatrix}. \tag{17}$$

The middle two equations then yields

$$\begin{pmatrix} C & 0 \\ 0 & D \end{pmatrix} \begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} 0 & -CB^T \\ DA^T & 0 \end{pmatrix} \begin{pmatrix} s^* \\ w^* \end{pmatrix} + \begin{pmatrix} 0 & -CD^T \\ DC^T & 0 \end{pmatrix} \begin{pmatrix} \lambda^* \\ \mu^* \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix}. \tag{18}$$

Because of the assumption $rank(C) = p = rank(D) = q$, the matrix DC^T is invertible, so is $\begin{pmatrix} 0 & -CD^T \\ DC^T & 0 \end{pmatrix}$, whose inverse is given by $\begin{pmatrix} 0 & (DC^T)^{-1} \\ -(CD^T)^{-1} & 0 \end{pmatrix}$. From (18) we have

$$\begin{pmatrix} \lambda^* \\ \mu^* \end{pmatrix} = -\begin{pmatrix} (DC^T)^{-1}DA^T & 0 \\ 0 & -(CD^T)^{-1}CB^T \end{pmatrix} \begin{pmatrix} s^* \\ w^* \end{pmatrix} + \begin{pmatrix} (DC^T)^{-1}d \\ -(CD^T)^{-1}c \end{pmatrix}.$$

Substituting this equation into (17) yields

$$\begin{aligned} \begin{pmatrix} x^* \\ y^* \end{pmatrix} = &\begin{pmatrix} 0 & -B^T + D^T(CD^T)^{-1}CB^T \\ A^T - C^T(DC^T)^{-1}DA^T & 0 \end{pmatrix} \begin{pmatrix} s^* \\ w^* \end{pmatrix} \\ &+ \begin{pmatrix} D^T(CD^T)^{-1}c \\ C^T(DC^T)^{-1}d \end{pmatrix}. \end{aligned} \tag{19}$$

Substituting this equation into the last two equations in (14) gives

$$(z^*)^T(z - Nz^* - b) \geq 0 \quad \forall z \in K,$$

where the notations are defined in (16). Then the optimality conditions in Theorem 4 is equivalent to finding z^* such that it satisfies $Nz^* + b \in K$ and the above inequality, which is a special case of GLVI (2). Since N defined in (16) is skew-symmetric, Theorem 5 follows from Lemma 1.

Again, the state vector z of (15) does not contain the variables of ELP (i.e., x and y). The relationship between the optimum (x^*, y^*) of problem (13) and the equilibrium point (s^*, w^*) is expressed in (19).

Remark 1. The assumption $p = q$ in Theorem 5 does not impose any strict requirement to the ELP problem (13) that can be solved by neural network (15). For example, if $p \neq q$, one can select $\min\{p, q\}$ equality constraints in $Cx = c$ and $Dy = d$, respectively; while putting the rest equality constraints into inequality constraints as discussed in the beginning of this subsection.

Remark 2. The dimensionality of the neural network in (15) is the same as that of the neural network in (12), though the ELP problem (13) has additional $2p$ equality constraints compared with the ELP problem (10). In other words, the equality constraints are "absorbed" by the neural network in (15). The premise is that there exist no bound constraints in the ELP problem. One may wonder if the equality constraints can still be "absorbed" by some neural network when the bound constraints are present. The answer is yes, because in this case, the bound constraints can be expressed as inequality constraints, as we pointed out earlier.

4 Numerical Examples

Example 1. Consider the following $\underline{\xi}$-norm minimization problem

$$\min \|x\|_1, \quad \text{s.t. } Ax \in \Omega_1,$$

where $x \in R^n, A \in R^{h\times n}, \Omega_1 = \{\xi \in R^h | \underline{\xi} \leq \xi \leq \overline{\xi}\}$ and $\|x\|_1 = \sum_{i=1}^n |x_i|$. It is shown in [7] that this problem is equivalent to the following ELP problem

$$\min_x \left\{ \begin{matrix} \max y^T x \\ y \in \mathcal{C} \end{matrix} \right\}, \quad \text{s.t.} Ax \in \Omega_1,$$

where $\mathcal{C} = \{y \in R^n | -e \leq y \leq e\}$ and $e = (1, 1, ..., 1)^T \in R^n$, which can be viewed as a special case of (10) with $B = I, \Omega_2 = \mathcal{C}$. Based on this observation, the neural network in (12) can be used to solve the problem. Specifically, the dynamic equation of the neural network becomes

$$\frac{d}{dt}\begin{pmatrix} s \\ w \end{pmatrix} = \Lambda \begin{pmatrix} I & -A \\ A^T & I \end{pmatrix} \begin{pmatrix} P_{\Omega_1}(-s - Aw) + Aw \\ P_{\Omega_2}(A^T s - w) - A^T s \end{pmatrix}. \tag{20}$$

It is interesting to note that this system shares a similar structure with system proposed in [7]:

$$\frac{d}{dt}\begin{pmatrix} x \\ s \end{pmatrix} = \begin{pmatrix} A^T & -I \\ I & A \end{pmatrix} \begin{pmatrix} P_{\Omega_1}(Ax - s) - Ax \\ P_{\Omega_2}(x + A^T s) - A^T s \end{pmatrix}.$$

Moreover, both systems are of $n + h$ dimensions. Let

$$A = \begin{pmatrix} 2 & 1 & 1 & 2 & 3 \\ 5 & 1 & -2 & 1 & 4 \\ 2 & 4 & -5 & 6 & -3 \end{pmatrix}, \underline{\xi} = \overline{\xi} = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix},$$

and use neural network (20) to solve the problem. All simulations with different initial points converge to the unique equilibrium point. Fig. 1 illustrates one of the simulation results with $\Lambda = I$. The corresponding solution to the problem is thus $x^* = (0.048, 0, -0.524, -0.286, 0)$, the same as that obtained in [7].

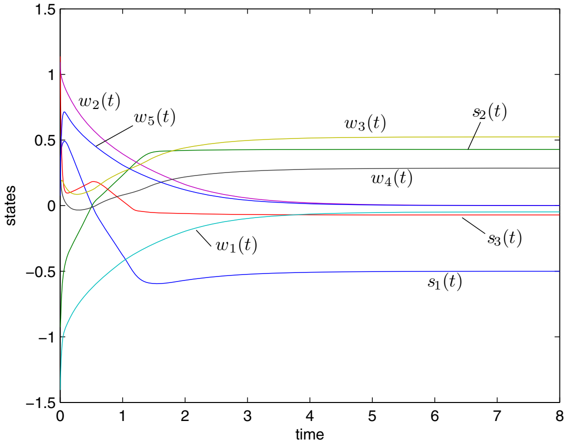


Fig. 1. Transient states of the neural network (20) with a random initial point in Example 1

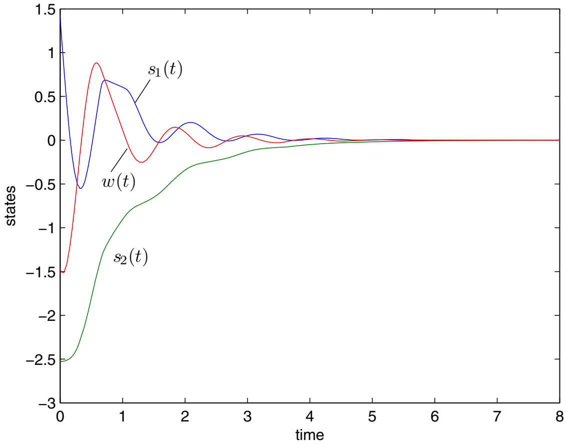


Fig. 2. Transient states of the neural network (15) with a random initial point in Example 2

Example 2. Consider an ELP problem with equality constraints in (13) with

$$A=\begin{pmatrix}-2 & 1 & 0 & 4\\ 1 & 0 & -1 & 2\end{pmatrix}, B=\begin{pmatrix}2 & 0 & 0 & -2\end{pmatrix}, C=\begin{pmatrix}0 & 1 & -1 & 1\\ 2 & 0 & 3 & 3\end{pmatrix}, D=\begin{pmatrix}1 & -2 & 1 & 0\\ 0 & 0 & 3 & 2\end{pmatrix},$$
$$\underline{\xi}=(0,0)^T, \overline{\xi}=(\infty,\infty)^T, \underline{\eta}=0, \overline{\eta}=\infty, c=d=(5,5)^T.$$

The exact solution is $x^*=(-2,4,1,2)^T, y^*=(0.5,-1.25,2,-0.5)^T$. We use neural network (15) to solve the problem. The states of the neural network $(s_1(t),s_2(t),w(t))^T$ always converge to the unique equilibrium point $(0,0,0)^T$

with any initial state, which corresponds to the exact solution of the ELP problem. Fig. 2 displays the transient behavior of the neural network with a random initial point and $\Lambda = I$. Moreover, by viewing the equality constraints as inequality constraints we have examined the performance of neural network (12) on this problem. Same solution has been obtained; while the neural network in (12) in this case has 7 states, which is higher in comparison with the neural network in (15) with only 3 states.

5 Concluding Remarks

In this paper, recurrent neural networks for solving general *extended linear programming (ELP)* problems are investigated. By transforming the optimality conditions for the problem with different types of constraints into different general linear variational inequalities (GLVI), three recurrent neural networks are developed for solving the corresponding problems based on a general neural network for solving GLVI. All of the neural networks are globally convergent to the solutions of the ELP problems under some mild conditions. Finally, two numerical examples are discussed to illustrate the performance of the neural networks.

References

1. Dantzig, G.B.: Linear Programming and Extensions. Princeton Univ. Press, Princeton (1980)
2. Maa, C.-Y., Shanblatt, M.A.: Linear and Quadratic Programming Neural Network Analysis. IEEE Trans. Neural Networks **3** (1992) 580–594
3. Wang, J.: A Deterministic Annealing Neural Network for Convex Programming. Neural Networks **7** (1994) 629–641
4. Xia, Y., Wang, J.: A Projection Neural Network and Its Application to Constrained Optimization Problems. IEEE Trans. Circuits Syst. I-Regul. Pap. **49** (2002) 447–458
5. Forti, M., Nistri, P., Quincampoix, M.: Generalized Neural Network for Nonsmooth Nonlinear Programming Problems. IEEE Trans. Circuits Syst. I-Regul. Pap. **51** (2004) 1741–1754
6. He, B.: Solution and Applications of a Class of General Linear Variational Inequalties. Science in China, Ser. A **39** (1996) 397–404
7. Xia, Y.: Neural Network for Solving Extended Linear Programming Problems. IEEE Trans. Neural Networks **8** (1997) 803–806
8. Gao, X.: A Neural Network for a Class of Extended Linear Variational Inequalities. Chinese Journal of Electronics **10** (2001) 471–475
9. Kinderlehrer, D., Stampcchia, G: An Introduction to Variational Inequalities and Their Applications. Academic Press, New York (1980)

A Recurrent Neural Network for Non-smooth Convex Programming Subject to Linear Equality and Bound Constraints

Qingshan Liu and Jun Wang

Department of Automation and Computer-Aided Engineering
The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong
{qsliu, jwang}@acae.cuhk.edu.hk

Abstract. In this paper, a recurrent neural network model is proposed for solving non-smooth convex programming problems, which is a natural extension of the previous neural networks. By using the non-smooth analysis and the theory of differential inclusions, the global convergence of the equilibrium is analyzed and proved. One simulation example shows the convergence of the presented neural network.

1 Introduction

In this paper, we are concerned with the following nonlinear programming problem:

$$(NP) \quad \begin{array}{ll} \text{minimize} & f(x), \\ \text{subject to} & Ax = b, \quad x \in \Omega, \end{array} \tag{1}$$

where $f(x) : R^n \to R$ is convex function but not necessary smooth, $A \in R^{m\times n}$, $b \in R^m$, and $\Omega \subset R^n$ is a bounded closed convex set.

Nonlinear programming has many applications in scientific and engineering problems, such as optimal control, signal and image processing, pattern recognition. In the past two decades, neural networks for optimization and their engineering applications have been widely investigated [1]-[13]. Tank and Hopfield [1] first proposed a neural network for solving linear programming problems, which motivated research and applications of neural networks for scientific and engineering problems. Kennedy and Chua [2] presented a neural network for solving nonlinear programming problems by utilizing the penalty parameter method. Zhang and Constantinides [3] proposed the Lagrangian network for solving nonlinear programming problems with equality constraints. Xia et al. [6] proposed a projection neural network with global convergence for solving nonlinear programming problems with convex set constraints. In particular, Forti et al. [10] proposed a generalized neural network for solving non-smooth nonlinear programming problems based on the gradient method and there is a parameter in that neural network which must be estimated beforehand. Gao [12] presented a recurrent neural network for solving the nonlinear convex programming problems by using the projection method. Recently, Li et al. [13] extended the projection neural network for solving the non-smooth convex optimization. In this paper,

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1004–1013, 2006.

a recurrent neural network is proposed for solving the non-smooth convex programming problems with equality and bound constraints based on the saddle point theorem and the global convergence of the neural network is analyzed and proved.

2 Preliminaries

Definition 1 [10]**:** *Supposing that for each point x in a set $E \subset R^n$ there corresponds a nonempty set $F(x) \subset R^n$, $x \rightsquigarrow F(x)$ is a set-valued map from E to R^n. A set-valued map $F : E \rightsquigarrow R^n$ with nonempty values is said to be upper semicontinuous at $x_0 \in E$ if for any open set V containing $F(x_0)$, there exists a neighborhood U of x_0 such that $F(U) \subset V$. Assume that E is closed, F has nonempty closed range, and it is bounded in a neighborhood of each point $x \in E$, then E is upper semicontinuous on E if and only if its graph $\{(x, y) \in E \times R^n : y \in F(x)\}$ is closed.*

Definition 2 [14]**:** *Let $V(x)$ be a function from R^n to R. For any $x \in R^n$,*

$$DV(x)(v) = \lim_{h \to 0+} \frac{V(x + hv) - V(x)}{h}.$$

We say that $DV(x)(v)$ is the derivative from the right of V at x in the direction v. If $DV(x)(v)$ exists for all directions, we say that V is differentiable from the right at x. We say that the closed convex subset (possibly empty)

$$\partial V(x) = \{p \in R^n : \forall v \in R^n, (p, v) \leq DV(x)(v)\}$$

is the sub-differential of V at x, where $(\cdot, \cdot)$ denotes the inner product in R^n. The element p of $\partial V(x)$ is called the sub-gradient of V at x.

The following property holds for the sub-differential of V at x.

Lemma 1 [15]**:** *Suppose that $V(x)$ is a convex function from R^n to R. The following result holds*

$$\partial V(x) = \{p \in R^n : V(x + y) - V(x) \geq (p, y), \forall y \in R^n\}.$$

In [12], when $f(x)$ in problem (1) is continuously differentiable, the following neural network is proposed for solving problem (1):

$$\begin{cases} \frac{dx}{dt} = 2\{-x + P_\Omega(x - \nabla f(x) + A^T(y - Ax + b))\}, \\ \frac{dy}{dt} = -Ax + b, \end{cases} \tag{2}$$

where $P_\Omega(u) : R^n \to \Omega$ is a projection operator defined by

$$P_\Omega(u) = \arg\min_{v \in \Omega} \|u - v\|.$$

Lemma 2 [16]**:** *For the Projection operator $P_\Omega(x)$, the following inequality holds:*

$$(v - P_\Omega(v))^T (P_\Omega(v) - u) \geq 0, \quad v \in R^n, u \in \Omega.$$

3 Model Description

In problem (1), when $f(x)$ is not smooth, we propose the following neural network described by differential inclusions and differential equations

$$\begin{cases} \frac{dx}{dt} \in 2\lambda\{-x + P_\Omega(x - \alpha(\partial f(x) - A^T(y - Ax + b)))\}, \\ \frac{dy}{dt} = \lambda(-Ax + b), \end{cases} \tag{3}$$

where $\partial f(x)$ is the sub-differential of $f(x)$ at x, λ and α are positive constants.

Definition 3. $[x^*, y^*]^T$ *is said to be an equilibrium of system* (3) *if there exists* $\gamma^* \in \partial f(x^*)$ *such that*

$$\begin{cases} -x^* + P_\Omega(x^* - \alpha(\gamma^* - A^T y^*)) = 0, \\ -Ax^* + b = 0. \end{cases} \tag{4}$$

We describe the relationship between the optimal solutions of problem (1) and the equilibrium points of system (3) as following theorem.

Theorem 1. *For any positive constants* α *and* β*,* $\Omega^* = \Omega^e_x$*, where* Ω^* *is the optimal solution set of problem* (1)*,* $\Omega^e_x = \{x : [x, y]^T \in \Omega^e\}$ *and* Ω^e *is the equilibrium point set of system* (3)*.*

Proof. The Lagrange function of problem (1) is

$$L(x, y) = f(x) - y^T(Ax - b), \tag{5}$$

where $y \in R^m$ is the Lagrange multiply. According to the saddle point theorem [16], x^* is an optimal solution of problem (1), if and only if there exists y^*, such that $[x^*, y^*]^T$ is a saddle point of $L(x, y)$ on $\Omega \times R^m$. That is,

$$L(x^*, y) \le L(x^*, y^*) \le L(x, y^*), \quad \forall x \in \Omega, \forall y \in R^m.$$

If $[x^*, y^*]^T \in \Omega^e$ is an equilibrium of system (3), then there exists $\gamma^* \in \partial f(x^*)$ such that

$$P_\Omega(x^* - \alpha(\gamma^* - A^T y^*)) - x^* = 0, \tag{6}$$

and

$$Ax^* - b = 0. \tag{7}$$

Let $v = x^* - \alpha(\gamma^* - A^T y^*)$ and $u = x$, by Lemma 2, it follows that

$$[x^* - \alpha(\gamma^* - A^T y^*) - P_\Omega(x^* - \alpha(\gamma^* - A^T y^*))]^T [P_\Omega(x^* - \alpha(\gamma^* - A^T y^*)) - x] \ge 0.$$

By equation (6), we get that

$$(x - x^*)^T(\gamma^* - A^T y^*) \ge 0, \quad \forall x \in \Omega. \tag{8}$$

Since $\gamma^* \in \partial f(x^*)$, we have $\gamma^* - A^T y^* \in \partial_x L(x^*, y^*)$, where $\partial_x L(x^*, y^*)$ is the sub-gradient of $L(x, y)$ with respect to x at $[x^*, y^*]^T$. By Lemma 1, it follows that

$$L(x, y^*) - L(x^*, y^*) \ge (\gamma^* - A^T y^*, x - x^*) \ge 0, \quad \text{for any } x \in \Omega.$$

From (7), we get that $L(x^*, y^*) - L(x^*, y) = 0$. So, $[x^*, y^*]^T$ is a saddle point of the Lagrange function $L(x, y)$ and x^* is an optimal solution of problem (1). That is $x^* \in \Omega^*$, then $\Omega_x^e \subset \Omega^*$.

On the other hand, suppose that $x^o \in \Omega^*$ is an optimal solution of problem (1), then there exists y^o such that $[x^o, y^o]^T$ is a saddle point of Lagrange function $L(x, y)$, i.e.,

$$L(x^o, y) \leq L(x^o, y^o) \leq L(x, y^o), \quad \forall x \in \Omega, \forall y \in R^m. \tag{9}$$

We show that there exists $\gamma^o \in \partial f(x^o)$ such that for any $x \in \Omega$,

$$(x - x^o)^T(\gamma^o - A^T y^o) \geq 0. \tag{10}$$

Otherwise, for any $\gamma \in \partial f(x^o)$, there exists $\hat{x} \in \Omega$ such that

$$(\hat{x} - x^o)^T(\gamma - A^T y^o) < 0.$$

From Lemma 1, we get that $L(x^o, y^o) - L(\hat{x}, y^o) \geq (\gamma - A^T y^o, x^o - \hat{x})$, then $L(\hat{x}, y^o) - L(x^o, y^o) \leq (\gamma - A^T y^o, \hat{x} - x^o) < 0$, which contradicts the right inequality in (9). Consequently, the inequality in (10) holds.

Let $v = x^o - \alpha(\gamma^o - A^T y^o)$ and $u = x^o$, by Lemma 2, we get that

$$[x^o - \alpha(\gamma^o - A^T y^o) - P_\Omega(x^o - \alpha(\gamma^o - A^T y^o))]^T[P_\Omega(x^o - \alpha(\gamma^o - A^T y^o)) - x^o] \geq 0.$$

Then

$$\begin{aligned}&[x^o - P_\Omega(x^o - \alpha(\gamma^o - A^T y^o))]^T[P_\Omega(x^o - \alpha(\gamma^o - A^T y^o)) - x^o]\\ &\geq \alpha[\gamma^o - A^T y^o]^T[P_\Omega(x^o - \alpha(\gamma^o - A^T y^o)) - x^o],\end{aligned}$$

i.e.,

$$-\|x^o - P_\Omega(x^o - \alpha(\gamma^o - A^T y^o))\|^2 \geq \alpha[\gamma^o - A^T y^o]^T[P_\Omega(x^o - \alpha(\gamma^o - A^T y^o)) - x^o].$$

From (10), $[\gamma^o - A^T y^o]^T[P_\Omega(x^o - \alpha(\gamma^o - A^T y^o)) - x^o] \geq 0$, then $-\|x^o - P_\Omega(x^o - \alpha(\gamma^o - A^T y^o))\|^2 \geq 0$, it follows that

$$-\|x^o - P_\Omega(x^o - \alpha(\gamma^o - A^T y^o))\|^2 = 0,$$

then

$$x^o = P_\Omega(x^o - \alpha(\gamma^o - A^T y^o)). \tag{11}$$

From the left inequality in (9), we get that

$$(y - y^o)^T(Ax^o - b) \geq 0, \quad \forall y \in R^m,$$

so

$$Ax^o - b = 0. \tag{12}$$

From Definition 3, $[x^o, y^o]^T$ is an equilibrium of system (3), then $\Omega^* \subset \Omega_x^e$.

From above proof, we get that $\Omega^* = \Omega_x^e$. This completes the proof. □

4 Global Convergence Analysis

In this section, we analyze and prove the global convergence of the proposed neural network (3) based on the non-smooth analysis and the theory of differential inclusions.

Lemma 3. *For any $\gamma \in \partial f(x)$, $x \in R^n$, $y \in R^m$, the following inequality holds:*

$$[x - P_\Omega(x)]^T [P_\Omega(x - \alpha(\gamma - A^T(y - Ax + b))) - x] \leq -\|x - P_\Omega(x)\|^2. \quad (13)$$

Proof.

$$\begin{aligned} & [x - P_\Omega(x)]^T [P_\Omega(x - \alpha(\gamma - A^T(y - Ax + b))) - x] \\ = & [x - P_\Omega(x)]^T [-x + P_\Omega(x) - P_\Omega(x) + P_\Omega(x - \alpha(\gamma - A^T(y - Ax + b)))] \\ = & -\|x - P_\Omega(x)\|^2 - [x - P_\Omega(x)]^T [P_\Omega(x) - P_\Omega(x - \alpha(\gamma - A^T(y - Ax + b)))]. \end{aligned}$$

From Lemma 2, we have $[x-P_\Omega(x)]^T[P_\Omega(x)-P_\Omega(x-\alpha(\gamma-A^T(y-Ax+b)))] \geq 0$, then the inequality in (13) holds. □

Theorem 2. *For system* (3)*, the solution $x(t)$ converges exponentially to the set Ω when the initial point $x_0 \notin \Omega$. Moreover, $x(t) \subset \Omega$ when $x_0 \in \Omega$.*

Proof. Let $g(x) = \|x - P_\Omega(x)\|^2$, then $g(x)$ is differentiable with respect to t. We get that

$$\begin{aligned} \frac{dg(x(t))}{dt} &= \left(\frac{dg(x)}{dx}\right)^T \left(\frac{dx}{dt}\right) \\ &\leq \sup_{\gamma \in \partial f(x)} 4\lambda [x - P_\Omega(x)]^T [P_\Omega(x - \alpha(\gamma - A^T(y - Ax + b))) - x]. \end{aligned}$$

By Lemma 3, we have

$$\frac{dg(x(t))}{dt} \leq -4\lambda \|x - P_\Omega(x)\|^2 = -4\lambda g(x).$$

Hence,

$$\|x - P_\Omega(x)\| \leq \|x_0 - P_\Omega(x_0)\| \exp(-2\lambda(t - t_0)).$$

When $x_0 \notin \Omega$, any solution $x(t)$ of system (3) converges exponentially to the feasible set Ω.

When $x_0 \in \Omega$, we have $x - P_\Omega(x) = 0$, i.e., $x(t) \subset \Omega$. □

Remark. From Theorem 2, we know that $\Omega \times R^m$ is a positive invariant and attractive set of system (3).

Theorem 3. *The state trajectory of the neural network* (3) *converges to the optimal solution set Ω^* of problem* (1)*.*

Proof. Let x^* be an optimal solution of problem (1), then, according to Theorem 1, there exist $y^* \in R^m$ and $\gamma^* \in \partial f(x^*)$ such that the equations in (4) hold. Construct an energy function as follows

$$\begin{aligned}E(x(t), y(t)) = \alpha\{f(x) + \frac{1}{2}\|y - Ax + b\|^2 - f(x^*) - \frac{1}{2}\|y^*\|^2 - (x - x^*)^T \\ \cdot(\gamma^* - A^T y^*) - (y - y^*)^T y^* + \frac{1}{2}\|y - y^*\|^2\} + \frac{1}{2}\|x - x^*\|^2.\end{aligned}$$

The sub-gradient of $E(x, y)$ with respect to x is

$$\partial_x E(x, y) = \alpha(\partial f(x) - A^T(y - Ax + b) - \gamma^* + A^T y^*) + x - x^*.$$

The gradient of $E(x, y)$ with respect to y is

$$\nabla_y E(x, y) = 2\alpha(y - y^*) - \alpha(Ax - b).$$

From the chain rule [17], the derivative of $E(x, y)$ along the solution of system (3) is

$$\begin{aligned}\dot{E}(x(t), y(t)) = [\alpha(\gamma - A^T(y - Ax + b) - \gamma^* + A^T y^*) + x - x^*]^T \dot{x}(t) \\ +[2\alpha(y - y^*) - \alpha(Ax - b)]^T \dot{y}(t), \quad \forall \gamma \in \partial f(x). \qquad (14)\end{aligned}$$

Then

$$\begin{aligned}\dot{E}(x(t), y(t)) \leq \sup_{\gamma \in \partial f(x)} 2\lambda[\alpha(\gamma - A^T(y - Ax + b) - \gamma^* + A^T y^*) + x - x^*]^T[\tilde{x} - x] \\ +\lambda[2\alpha(y - y^*) - \alpha(Ax - b)]^T[-Ax + b], \qquad (15)\end{aligned}$$

where $\tilde{x} = P_\Omega(x - \alpha(\gamma - A^T(y - Ax + b)))$.

We first prove the following equality

$$\begin{aligned}&2[\alpha(\gamma - A^T(y - Ax + b) - \gamma^* + A^T y^*) + x - x^*]^T[\tilde{x} - x] \\ &+[2\alpha(y - y^*) - \alpha(Ax - b)]^T[-Ax + b] \\ =\ &2[\alpha(\gamma - A^T(y - Ax + b) - \gamma^* + A^T y^*) + x - x^*]^T[\tilde{x} - x^*] \\ &+2[\alpha(\gamma - A^T(y - Ax + b) - \gamma^* + A^T y^*) + x - x^*]^T[x^* - x] \\ &+2\alpha[y - y^*]^T[-Ax + b] + \alpha\|Ax - b\|^2 \\ =\ &2[\alpha(\gamma - A^T(y - Ax + b) - \gamma^* + A^T y^*) - x + \tilde{x}]^T[\tilde{x} - x^*] \\ &+2[x - \tilde{x} + x - x^*]^T[\tilde{x} - x^*] + 2\alpha[\gamma - \gamma^*]^T[x^* - x] \\ &+2[\alpha(-A^T(y - Ax + b) + A^T y^*) + x - x^*]^T[x^* - x] \\ &+2\alpha[y - y^*]^T[-Ax + b] + \alpha\|Ax - b\|^2 \\ =\ &-2[x - \alpha(\gamma - A^T(y - Ax + b)) - \tilde{x}]^T[\tilde{x} - x^*] - 2\alpha[\gamma^* - A^T y^*]^T[\tilde{x} - x^*] \\ &+2[x - \tilde{x} + x - x^*]^T[\tilde{x} - x^*] + 2\alpha[\gamma - \gamma^*]^T[x^* - x] \\ &+2\alpha[-A^T(y - Ax + b) + A^T y^*]^T[x^* - x] + 2[x - x^*]^T[x^* - x] \\ &+2\alpha[y - y^*]^T[-Ax + b] + \alpha\|Ax - b\|^2 \\ =\ &-2[x - \alpha(\gamma - A^T(y - Ax + b)) - \tilde{x}]^T[\tilde{x} - x^*] - 2\alpha[\gamma^* - A^T y^*]^T[\tilde{x} - x^*] \\ &-2\|x - \tilde{x}\|^2 + 2\alpha[\gamma - \gamma^*]^T[x^* - x] - \alpha\|Ax - b\|^2. \qquad (16)\end{aligned}$$

From Lemma 2 and inequality (8), we have

$$[x-\alpha(\gamma-A^T(y-Ax+b))-\tilde{x}]^T[\tilde{x}-x^*]\geq 0 \tag{17}$$

and

$$(\gamma^*-A^Ty^*)^T(\tilde{x}-x^*)\geq 0. \tag{18}$$

Since $f(x)$ is convex, from Lemma 1, we have

$$f(x)-f(x^*)\geq {\gamma^*}^T(x-x^*),$$

and

$$f(x^*)-f(x)\geq \gamma^T(x^*-x),$$

then

$$(\gamma-\gamma^*)^T(x^*-x)\leq 0. \tag{19}$$

From (15)-(19), we get that

$$\begin{aligned}\dot{E}(x(t),y(t)) &\leq \lambda \sup_{\gamma\in\partial f(x)}\{-2\|x-P_\Omega(x-\alpha(\gamma-A^T(y-Ax+b)))\|^2\}\\ &\quad -\lambda\alpha\|Ax-b\|^2\\ &= -\lambda \inf_{\gamma\in\partial f(x)}\{2\|x-P_\Omega(x-\alpha(\gamma-A^T(y-Ax+b)))\|^2\}\\ &\quad -\lambda\alpha\|Ax-b\|^2. \end{aligned} \tag{20}$$

Let $\varphi(x,y)=f(x)+1/2\|y-Ax+b\|^2$, then, $\varphi(x,y)$ is also a convex function. According to Lemma 1, we have

$$\varphi(x,y)-\varphi(x^*,y^*)\geq (x-x^*)^T(\gamma^*-A^Ty^*)+(y-y^*)^Ty^*, \tag{21}$$

thus

$$E(x,y)\geq \frac{1}{2}\|x-x^*\|^2+\frac{\alpha}{2}\|y-y^*\|^2. \tag{22}$$

Let $G(x_0,y_0)=\{[x,y]^T : E(x,y)\leq E(x_0,y_0),[x,y]^T\in R^n\times R^m\}$, then for any initial point $[x_0,y_0]^T\in R^n\times R^m$, $G(x_0,y_0)$ is bounded. It follows that $[x(t),y(t)]^T$ is also bounded.

Define $D(x,y)=\inf_{\gamma\in\partial f(x)}\{2\|x-P_\Omega(x-\alpha(\gamma-A^T(y-Ax+b)))\|^2\}+\alpha\|Ax-b\|^2$. If $[\hat{x},\hat{y}]^T\in\Omega^e$, we have $\dot{E}(\hat{x}(t),\hat{y}(t))=0$. From (20), we get that $D(\hat{x},\hat{y})=0$. On the other hand, if there exists $[\check{x},\check{y}]\in\Omega\times R^m$ such that $D(\check{x},\check{y})=0$, since $f(x)$ is a convex function, by convex analysis, $\partial f(x)$ is a nonempty and compact convex subset on R^n, then there exists $\check{\gamma}\in\partial f(\check{x})$ such that

$$\check{x}-P_\Omega(\check{x}-\alpha(\check{\gamma}-A^T(\check{y}-A\check{x}+b)))=0,$$

and

$$A\check{x}-b=0.$$

Therefore, $D(x,y)=0$ if and only if $[x,y]^T\in\Omega^e$.

From the boundedness of $[x(t), y(t)]^T$ and Ω, we get that $\|\dot{x}(t)\| + \|\dot{y}(t)\|$ is also bounded, denoted by M. Then, there exists an increasing sequence $\{t_n\}$ with $\lim_{n\to\infty} t_n \to \infty$ and a limit point $[\bar{x}, \bar{y}]^T$ such that $\lim_{n\to\infty} x(t_n) \to \bar{x}$ and $\lim_{n\to\infty} y(t_n) \to \bar{y}$. Similarly to the proof in [13], we will prove that $D(\bar{x}, \bar{y}) = 0$. If it does not hold, that is $D(\bar{x}, \bar{y}) > 0$. Since $D(x, y)$ is lower semi-continuous and continuous with respect to x and y respectively, there exist $\delta > 0$ and $m > 0$, such that $D(x, y) > m$ for all $[x, y]^T \in B([\bar{x}, \bar{y}]^T, \delta)$, where $B([\bar{x}, \bar{y}]^T, \delta) = \{[x, y]^T : \|x - \bar{x}\| + \|y - \bar{y}\| \le \delta\}$ is the δ neighborhood of $[\bar{x}, \bar{y}]^T$. Since $\lim_{n\to\infty} x(t_n) \to \bar{x}$ and $\lim_{n\to\infty} y(t_n) \to \bar{y}$, there exists a positive integer N, such that for all $n \ge N$, $\|x(t_n) - \bar{x}\| + \|y(t_n) - \bar{y}\| < \delta/2$. When $t \in [t_n - \frac{\delta}{4M}, t_n + \frac{\delta}{4M}]$ and $n \ge N$, we have

$$\begin{aligned}\|x(t) - \bar{x}\| + \|y(t) - \bar{y}\| &\le \|x(t) - x(t_n)\| + \|y(t) - y(t_n)\| \\ &\quad + \|x(t_n) - \bar{x}\| + \|y(t_n) - \bar{y}\| \\ &\le M|t - t_n| + \frac{\delta}{2} \le \delta.\end{aligned}$$

It follows that $D(x, y) > m$ for all $t \in [t_n - \frac{\delta}{4M}, t_n + \frac{\delta}{4M}]$. Since the Lebesgue measure of the set $t \in \bigcup_{n\ge N}[t_n - \frac{\delta}{4M}, t_n + \frac{\delta}{4M}]$ is infinite, then we have

$$\int_0^\infty D(x(t), y(t))dt = \infty. \tag{23}$$

On the other hand, by (20), $E(x(t), y(t))$ is non-increasing and bounded, then, there exists a constant E_0 such that $\lim_{t\to\infty} E(x(t), y(t)) = E_0$. We have

$$\begin{aligned}\int_0^\infty D(x(t), y(t))dt &= \lim_{s\to\infty} \int_0^s D(x(t), y(t))dt \\ &\le - \lim_{s\to\infty} \frac{1}{\lambda} \int_0^s \dot{E}(x(t), y(t))dt \\ &= -\frac{1}{\lambda} \left[\lim_{s\to\infty} E(x(s), y(s)) - E(x(0), y(0)) \right] \\ &= -\frac{1}{\lambda} \left[E_0 - E(x(0), y(0)) \right],\end{aligned} \tag{24}$$

which contradicts (23). Therefore, we have that $D(\bar{x}, \bar{y}) = 0$, and then $[\bar{x}, \bar{y}]^T \in \Omega^e$. That is the limit point $[\bar{x}, \bar{y}]^T$ being an equilibrium of system (3).

Finally, we will prove that

$$\lim_{t\to+\infty} \operatorname{dist}([x(t), y(t)]^T, \Omega^e) = 0. \tag{25}$$

Otherwise, there exists a constant $\varepsilon > 0$ such that for any $T > 0$, there exists a $\hat{t} \ge T$ which satisfies $\operatorname{dist}([x(\hat{t}), y(\hat{t})]^T, \Omega^e) \ge \varepsilon$. By the boundedness property of $[x(t), y(t)]^T$, we can choose a convergent subsequence $\{[x(\hat{t}_m), y(\hat{t}_m)]^T\}$, which satisfies $\lim_{\hat{t}_m\to+\infty} x(\hat{t}_m) = \hat{x}$ and $\lim_{\hat{t}_m\to+\infty} y(\hat{t}_m) = \hat{y}$ with $[\hat{x}, \hat{y}]^T \in \Omega^e$, such that

$$\operatorname{dist}([x(\hat{t}_m), y(\hat{t}_m)]^T, \Omega^e) \ge \varepsilon, \quad (m = 1, 2, \ldots).$$

Letting $\hat{t}_m \to +\infty$, we have

$$\mathrm{dist}([\hat{x}, \hat{y}]^T, \Omega^e) \geq \varepsilon > 0,$$

which contracts $\mathrm{dist}([\hat{x}, \hat{y}]^T, \Omega^e) = 0$ since $[\hat{x}, \hat{y}]^T \in \Omega^e$. Then (25) holds. That is, for any initial point $[x_0, y_0]^T \in R^n \times R^m$, the trajectory $[x(t), y(t)]^T$ corresponding to system (3) converges to the equilibrium point set Ω^e, in which $x(t)$ converges to the optimal solution set Ω^* of problem (1). □

5 Simulation Example

Example. Consider the following non-smooth nonlinear programming problem

$$\begin{array}{ll} \text{minimize} & (x_1-1)^2 + |x_1+x_2| + |x_1-x_2|, \\ \text{subject to} & x_1 + 2x_2 = -1, \quad x \in \Omega = \{[x_1, x_2]^T : -5 \leq x_i \leq 5, i = 1, 2\}. \end{array} \tag{26}$$

For this problem, letting $\lambda = \alpha = 1$, the system (3) can be written as

$$\begin{cases} \frac{dx_1}{dt} \in 2\{P_\Omega(-2x_1 - 2x_2 - \Theta(x_1+x_2) - \Theta(x_1-x_2) + y + 1) - x_1\}, \\ \frac{dx_2}{dt} \in 2\{P_\Omega(-2x_1 - 3x_2 - \Theta(x_1+x_2) + \Theta(x_1-x_2) + 2y - 2) - x_2\}, \\ \frac{dy}{dt} = -x_1 - 2x_2 - 1, \end{cases} \tag{27}$$

where $\Theta(\cdot)$ is a bipolar activation function defined as

$$\Theta(\rho) = \begin{cases} 1, & \rho > 0, \\ [-1, 1], & \rho = 0, \\ -1, & \rho < 0. \end{cases} \tag{28}$$

Figure 1 shows that the simulation result by selecting 20 random initial points. We can see that all the state trajectories $[x_1(t), x_2(t)]^T$ converge to the optimal solution $[0.5, -0.75]^T$.

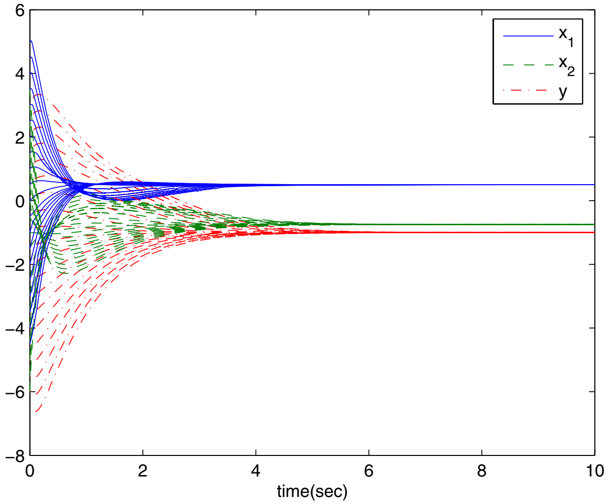


Fig. 1. Transient behavior of system (27) in Example

6 Conclusions

In this paper, we present a recurrent neural network model for solving non-smooth convex programming problems with linear equalities and bound constraints. The global convergence of the neural network is proven by using the non-smooth analysis and the theory of differential inclusions. One simulation example is given to illustrate the results in this paper.

References

1. Tank, D.W., Hopfield, J.J.: Simple Neural Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit. IEEE Trans. Circuits and Systems, **33** (1986) 533-541
2. Kennedy, M.P., Chua, L.O.: Neural Networks for Nonlinear Programming. IEEE Trans. Circuits and Systems, **35** (1988) 554-562
3. Zhang, S., Constantinides, A.G.: Lagrange Programming Neural Networks. IEEE Trans. Circuits and Systems II, **39** (1992) 441-452
4. Wang, J.: Analysis and Design of a Recurrent Neural Network for Linear Programming. IEEE Trans. Circuits and Systems I, **40** (1993) 613-618
5. Wang, J.: A Deterministic Annealing Neural Network for Convex Programming. Neural Networks, **7** (1994) 629-641
6. Xia, Y., Leung, H., Wang, J.: A Projection Neural Networks and its Application to Constrained Optimization Problems. IEEE Trans. Circuits Syst. II, **49** (2002) 447-458
7. Xia, Y., Wang, J.: A General Projection Neural Network for Solving Monotone Variational Inequalities and Related Optimization Problems. IEEE Trans. Neural Networks, **15** (2004) 318-328
8. Xia, Y., Wang, J.: A Recurrent Neural Network for Solving Nonlinear Convex Programs Subjected to Linear Constraints. IEEE Trans. Neural Networks, **16** (2005) 379-386
9. Forti, M., Nistri, P.: Global Convergence of Neural Networks with Discontinuous Neuron Activations. IEEE Trans. Circuis Syst. I, **50** (2003) 1421-1435
10. Forti, M., Nistri, P., Quincampoix, M.: Generalized Neural Network for Nonsmooth Nonlinear Progranmming Problems. IEEE Trans. Circuis Syst. I, **51** (2004) 1741-1754
11. Lu, W.L., Chen, T.P.: Dynamical Behaviors of Cohen-Grossberg Neural Networks with Discontinuous Activation Functions. Neural Networks, **18** (2005) 231-242
12. Gao, X.B.: A Novel Neural Network for Nonlinear Convex Programming. IEEE Trans. Neural Networks, **15** (2004) 613-621
13. Li, G., Song, S., Wu, C., Du, Z.: A Neural Network Model for Non-smooth Optimization over a Compact Convex Subset. In: Proceedings of ISNN 2006, LNCS 3971 (2006) 344-349
14. Aubin, J.P., Cellina, A.: Differential Inclusions. Berlin, Germany: Springer (1984)
15. Tuy, H.: Convex Analysis and Global Optimization. Kluwer, Netherlands (1998)
16. Kinderlehrer, D., Stampcchia, G.: An Introduction to Variational Inequalities and Their Applications. New York: Academic Press (1980)
17. Clarke, F.H.: Optimization and Non-smooth Analysis. New York: Wiley (1990)

Neural Networks for Optimization Problem with Nonlinear Constraints

Min-jae Kang[1], Ho-chan Kim[1], Farrukh Aslam Khan[2], Wang-cheol Song[2], and Sang-joon Lee[2]

[1] Department of Electrical & Electronic Engineering, Cheju National University,
Jeju 690-756, South Korea
{minjk, hckim}@cheju.ac.kr
[2] Department of Computer Engineering, Cheju National University,
Jeju 690-756, South Korea
{farrukh, philo, sjlee}@cheju.ac.kr

Abstract. Hopfield introduced the neural network for linear programming with linear constraints. In this paper, Hopfield neural network has been generalized to solve the optimization problems including nonlinear constraints. The proposed neural network can solve a nonlinear cost function with nonlinear constraints. Also, methods have been discussed to reconcile optimization problems with neural networks and implementation of the circuits. Simulation results show that the computational energy function converges to stable point by decreasing the cost function as the time passes.

1 Introduction

Many advantages of neural networks have been published because of their parallel processing characteristics and ability to find suitable solutions for various kinds of problems. On the other hand, these solutions could be of some disadvantage because neural networks sometimes can not find the best solution. Neural networks often trap to the local minimum and then it is difficult to escape from there. However, fast better decisions are often proved to be more important than slow best decisions in many cases of real life. Therefore, still many researchers are interested in neural networks even though local minimum problem exists.

Several papers have been published about neural networks for linear programming problems since the time Hopfield presented the simple linear programming neural networks. Maa and Shanblant [4] introduced versatile input-output characteristic functions of neuron to improve neural network performance, Huertas [1] used many different types of sources to implement Neural Networks, Chua and Lin [2] published about electronic nonlinear parts needed for neural networks and Kennedy and Chua [3] studied in specific when to implement neural networks and explained considerations for using electrical parts.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1014 – 1021, 2006.

In this paper, neural networks for nonlinear programming have been proposed. Kennedy and Chua published the nonlinear programming neural networks, which can handle a nonlinear cost function with linear constraint. The proposed neural network in this paper can solve a nonlinear cost function with nonlinear constraints. Pspice has been used for circuit level simulations. Simulation results show that these neural networks converge to stable point by decreasing the cost function.

2 Hopfield Linear Programming Networks

Linear programming problem can be defined as a method of minimizing the cost function, that is,

$$\phi(V) = AV \tag{1}$$

Here, V is an n-dimensional variable and A is constant. This function becomes minimized at the same time satisfying m constraints

$$W(V) = DV \geq B \tag{2}$$

D represents the coefficients of variables V, B is the constraint area. Fig. 1 shows electrical Hopfield model for linear programming problem in case of 2 variables and 4 constraints. This system converges to minimize cost function as time elapses.

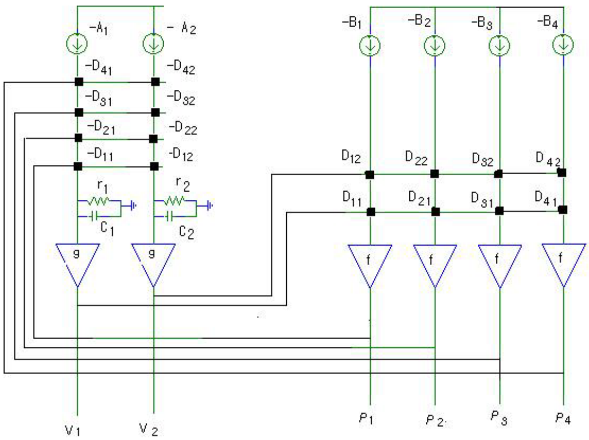


Fig. 1. Neural Network of 2-variable, 4 constraints linear programming

In Fig.1, operational amplifiers represent neurons. Two different types of input-output transfer functions have been used and a linear function for operational amplifier of variables are used as follows,

$$V = g(u) = ku \tag{3}$$

Here, k is a positive constant; u is an input of operational amplifier. A nonlinear function is used for operational amplifier of constraints as follows

$$P = f(z), \qquad z = DV - B$$
$$where \quad f(z) = \begin{pmatrix} 0, & z \geq 0 \\ z, & z < 0 \end{pmatrix} \tag{4}$$

Hopfield introduced the Lyapunov like computational energy function as follows [1].

$$E = AV + \sum_j F(D_j V - B_j) + \sum_{i=0}^{n-1} G_i \int_0^{V_i} g_i^{(-1)}(z)dz$$
$$here, \quad P = f(z) = \frac{dF(z)}{dz} \tag{5}$$

3 Neural Network for Non-linear Constraints

3.1 Stability for the Proposed Neural Network

By modifying the first term in equation (5), Kennedy and Chua proposed the nonlinear programming neural networks for nonlinear cost function with linear constraints [3]. In this paper, for the nonlinear constraints, we modified the second term in equation (5) as follows

$$E = \pi(V) + \sum_j H(w(V)) + \sum_{i=0}^{n-1} G_i \int_0^{V_i} g_i^{(-1)}(z)dz \tag{6}$$

Here, $\pi(V)$, $w(V)$ are nonlinear functions. The first and second terms in equation (6) represent cost function and constraint functions respectively. The third term is for the system's stable convergence. To write this computational energy function in the form of Lyapunov function, the time differential of this function should be negative or zero, that is,

$$\frac{dE}{dt} \leq 0 \tag{7}$$

Using chain rule and equation (3), equation (7) can be written as follows

$$\frac{dE}{dt} = \frac{dE}{dV}\frac{dV}{dt} = k\frac{dE}{dV}\frac{du}{dt} \leq 0 \tag{8}$$

If the capacitor's outcoming current at neuron's input node is made same as the computational energy function's differential, we can write

$$\frac{dE}{dV} = -C\frac{du}{dt} \tag{9}$$

And if k is positive, then the time differential of computational energy function is always negative or zero. Therefore, this system converges to stable point by decreasing the computational energy function. The outcoming current of capacitor is as follows

$$C\frac{du}{dt} = -\frac{dE}{dV} = -\frac{\pi(V)}{dV} - \sum\frac{dw}{dV}\frac{dH(w)}{dw} - Gu \tag{10}$$

3.2 Implementing the Constraint Function

The implementing method of nonlinear cost function has been showed in Kennedy and Chua's paper [3]. Therefore, in this part, we will show only how to implement the nonlinear constraints.

The computational energy function should be positive when constraints are not satisfied. Further more, this energy function should increase sharply as the system moves out of constraint boundary. The computational energy function satisfying all above conditions can be described as follows:

$$E_2(V) = \sum H(w(V)) = \sum\int_0^w h(w)dw = \begin{Bmatrix} 0, & w(V) \geq 0 \\ positive, & w(V) < 0 \end{Bmatrix} \tag{11}$$

It is required to find function h satisfying equation (11). One of those functions is as follows:

$$h(x) = \begin{Bmatrix} 0, & x \geq 0 \\ kx, & x < 0 \end{Bmatrix} \tag{12}$$

The integration of this function can be computed as,

$$H(z) = \int_0^x kxdx = \begin{Bmatrix} 0, & x \geq 0 \\ \frac{k}{2}x^2, & x < 0 \end{Bmatrix} \tag{13}$$

As in linear programming neural networks, each neuron is needed for each constraint, so the incoming current of neuron is as follows:

$$I_p = w_p(V) \qquad ,w_p(V) \geq 0 \tag{14}$$

Incoming current of the constraint neuron should be same as the negative direction of differential of energy function,

$$C\frac{du}{dt}=-\frac{dE}{dV}=-\sum\frac{dH}{dw}\frac{dw}{dV} \tag{15}$$

Therefore, time differential of energy function can be negative or zero.

Using equation (11), equation (15) can be re-written as follows:

$$C\frac{du}{dt}=-\frac{dE}{dV}=-\sum\ \frac{dw}{dV}h \tag{16}$$

4 Simulation

The following function is selected for simulation of nonlinear programming problem

$$f(x,y)=x^2+y^2 \tag{17}$$

And next two functions are used for constraints

$$\begin{aligned} g_1(x,y) &= x^2 - y \ \ \geq 0 \\ g_2(x,y) &= x \ + y \ -2 \geq 0 \end{aligned} \tag{18}$$

As seen in Fig. 2, these equations present the boundary of constraints in xy plane. By mapping x, y variables to the variable neuron outputs V$_1$, V$_2$, the incoming currents to the constraint neurons can be obtained as follows

$$\begin{aligned} I_1(V) &= V_1^2 - V_2 \\ I_2(V) &= V_1 + V_2 - 2 \end{aligned} \tag{19}$$

Pspice is used for circuit level simulation. Schematic circuit for simulation is shown in Fig. 2. Resistors are used for connections (D_{ji}) and resistor values are reciprocal of connection values. Also voltage sources and resistors are used for representing incoming currents (B_j). In this circuit, the voltage source (1V) and the resistor (-0.5Ω) are used for the current source (-2A). This is because the OP-AMP's input can be regarded as a virtual ground. By using equation (10), the incoming current to variable neurons' inputs can be obtained as follows

$$\begin{bmatrix} c_1\dfrac{du_1}{dt} \\ c_2\dfrac{du_2}{dt} \end{bmatrix} = \begin{bmatrix} -2V_1 \\ -2V_2 \end{bmatrix} + \begin{bmatrix} -2h_1V_1 - h_2 \\ h_1 - 2h_2V_2 \end{bmatrix} = \begin{bmatrix} -2V_1 - 2h_1V_1 - h_2 \\ -2V_2 + h_1 - 2h_2V_2 \end{bmatrix} \tag{20}$$

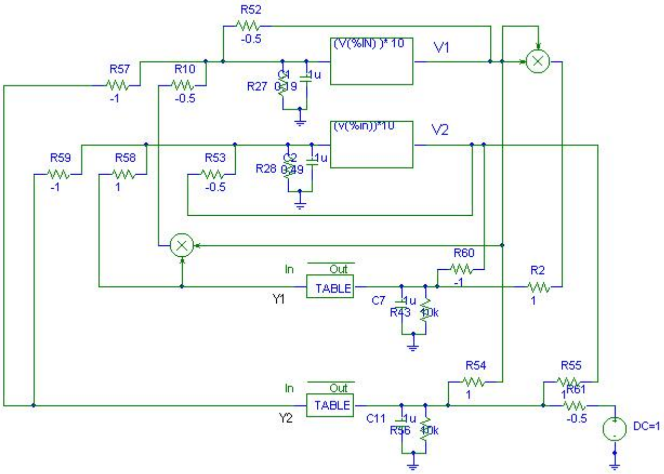


Fig. 2. Schematic diagram of the simulation circuit

ABM(Analog Behavior Modeling) are used to implement neurons and for multiplying variables as seen in Fig. 2. The parasitic resistor and capacitor at variable neurons' input are necessary for the stable system. The convergence speed of system depends on capacitor's size. Parasitic resistors' values should be selected carefully because the system's performance is affected by these resistors.

The contour lines of cost function and satisfying constraints area are shown in Fig. 3. The global minimum is located at (0,0), however the constrained minimum is at (1,1) as shown in Fig. 3.

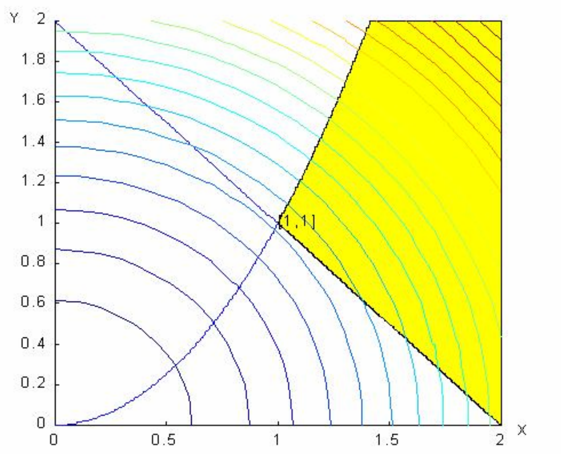


Fig. 3. The contour of cost function and constrained region

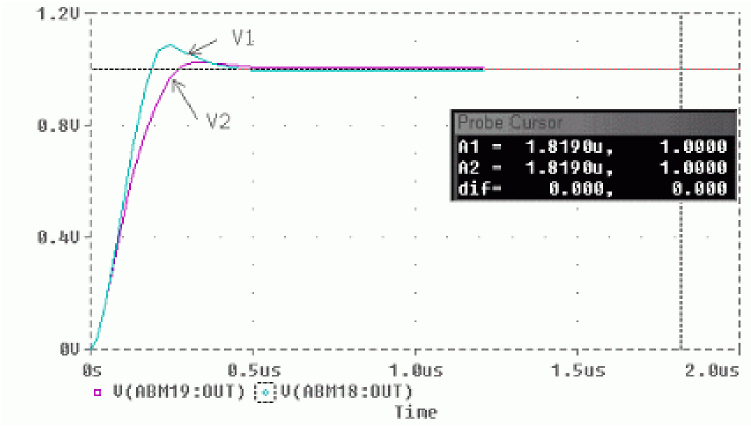


Fig. 4. Simulation result using Pspice

The Pspice transient simulation is shown in Fig. 4. The two variables V_1 and V_2 converge to the minimum point (1,1) after about 0.5μs.

5 Conclusion

Many papers for linear programming problem using neural networks have been published since Hopfield mentioned the possibility that his neural networks could be used for optimization problem. In this paper, the neural network for optimization problem with nonlinear constraint is proposed by extending the concept of a linear programming neural network. Therefore, the proposed neural network can solve linear programming problems as well as nonlinear programming problems.

Lyapunov function like computational energy is regarded as the first consideration factor for implementing neural networks for nonlinear programming problem. The computational energy function should be converged to stable point by decreasing its energy function as time passes. It has been analyzed that the incoming current to neuron is same as the negative direction differential of computational energy in order to satisfy the system for Lyapunov condition. The next consideration factor for implementing neural networks is an input-output transfer function of neuron. Two different types of neuron transfer functions are used, that is, one for variable neuron and another for constraint neuron. Both transfer functions should be designed to decrease computational energy function as the neural networks converge to the constrained minimum.

Pspice simulation program is used for circuit level simulation, resistor is used for synapses, and voltage sources and resistors are able to be used for current sources because neuron inputs can be regarded as virtual grounds. The transient simulation result shows that the neural network for nonlinear programming problem converges to a minimum after about 0.5μs for simple test problems.

Acknowledgement

The part of researchers participating in this study is supported by the grant from “the 2nd phase BK21 project”.

References

[1] J.L. Huertas and A. Rueda, "Synthesis of resistive n-port section-wise piecewise-linear networks," IEEE Trans. Circuits and Syst., vol. CAS-29, pp.6-14, Jan. 1982.

[2] L. O. Chua and G. N. Lin, "Nonlinear programming without computation," IEEE Trans. Circuits Syst., vol. CAS-32, pp. 736-742, July 1985.

[3] Kennedy and Chua, "Neural Networks for Nonlinear Programming", IEEE Trans. On Circuit and Systems, Vol. 35, pp. 554-562. 1988

[4] C. Y. Maa and M. Shanblatt, "Improved Linear Programming Neural Networks," IEEE Int. Conf. on Neural Networks, vol. 3, pp.748-751, 1989.

[5] Yee Leung, Kai-Zhou Chen, and Xing-Bao Gao, "A high-performance feedback neural network for solving convex Nolinear Programming Problem ", IEEE Trans. On Neural Networks, Vol. 9, pp. 1331-1343. 1998

[6] Xue-Bin, Jun Wang, "A Recurrent Neural Networks for Nonlinear Optimization with A Continuously Differentiable Objective Function with Bound Constraints ", IEEE Trans. On Neural Networks, Vol. 11, pp. 1251-1262. 2000

[7] Y.S. Xia and J. Wang, "On the stability of globally projected dynamical systems", J. Optimizat. Theory Applicat., Vol. 106, pp. 129-160. 2000

[8] Sabri Arik "An Analysis of global Asymtotic Stability for Cellur Delayed Neural Networks ", IEEE Trans. On Neural Networks, Vol. 13, pp. 1239-1342. 2002

[9] Xing-Bao Gao "A Novel Neural Network for Nonlinear Convex Programming", IEEE Trans. On Neural Networks, Vol. 11, pp. 613-621. 2004

A Novel Chaotic Annealing Recurrent Neural Network for Multi-parameters Extremum Seeking Algorithm

Yun-an Hu, Bin Zuo, and Jing Li

Department of Control Engineering, Naval Aeronautical Engineering Academy
Yan tai 264001, P.R. China
hya507@yahoo.com, zuobin97117@163.com

Abstract. The application of sinusoidal periodic search signals into the general extremum seeking algorithm(ESA) results in the "chatter" problem of the output and the switching of the control law and incapability of escaping from the local minima. A novel chaotic annealing recurrent neural network (CARNN) is proposed for ESA to solve those problems in the general ESA and improve the capability of global searching. The paper converts ESA into seeking the global extreme point where the slope of Cost Function is zero, and applies a CARNN to finding the global point and stabilizing the plant at that point. ESA combined with CARNN doesn't make use of search signals such as sinusoidal periodic signals, which solves those problems in previous ESA and improves the dynamic performance of the ESA system greatly. During the process of optimization, chaotic annealing is realized by decaying the amplitude of the chaos noise and the probability of accepting continuously. The process of optimization was divided into two phases: the coarse search based on chaos and the elaborate search based on RNN. At last, CARNN will stabilize the system to the global extreme point. At the same time, it can be simplified by the proposed method to analyze the stability of ESA. The simulation results of a simplified UAV tight formation flight model and a typical testing function proved the advantages mentioned above.

Keywords: Recurrent Neural Network, Extremum Seeking Algorithm, Annealing, Chaos, UAV.

1 Introduction

Early work on performance improvement by extremum seeking can be found in Tsien. In the 1950s and 1960s, ESA was considered as an adaptive control method [1]. Until 1990s sliding mode control for extremum seeking has not been utilized successfully [2]. Subsequently, a method of adding compensator dynamics in ESA was proposed by Krstic, which improved the stability of the system [3]. Although those methods improved tremendously the performance of ESA, the "chatter" problem of the output and the switching of the control law and incapability of escaping from the local minima limit the application of ESA.

The method of combining a chaotic annealing recurrent neural network with ESA is proposed in the paper. First, this paper converts ESA into seeking the global

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1022 – 1031, 2006.

extreme point where the slope of cost function is zero. Second, constructs a CARNN; then, applies the CARNN to finding the global extreme point and stabilizing the plant at that point. The CARNN proposed in the paper doesn't make use of search signals such as sinusoidal periodic signals, so the method can solve the "chatter" problem of the output and the switching of the control law in the general ESA and improve the dynamic performance of the ESA system. At the same time, CARNN utilizes the randomicity and the property of global searching of chaos system to improve the capability of global searching of the system [4,5], During the process of optimization, chaotic annealing is realized by decaying the amplitude of the chaos noise and the accepting probability continuously. Adjusting the probability of acceptance could influence the rate of convergence. The process of optimization was divided into two phases: the coarse search based on chaos and the elaborate search based on RNN. At last, CARNN will stabilize the system to the global extreme point, which is validated by simulating a simplified UAV tight formation flight model and a typical testing function. At the same time, it can be simplified by the proposed method to analyze the stability of ESA.

2 Problem Formulation

Consider a general nonlinear system:

$$\begin{aligned} \dot{x} &= f\left(x(t),u(t)\right) \\ y &= F\left(x(t)\right) \end{aligned} \tag{1}$$

Where $x \in R^n, u \in R^m$ and $y \in R$ are the states, the system inputs and the system output, respectively. $F\left(x\right)$ is also defined as the cost function of the system. $f\left(x,u\right)$ and $F\left(x\right)$ are smooth functions [3].

Assumption 1: There is a smooth control law:

$$u(t) = \alpha\left(x(t),\theta\right) \tag{2}$$

to stabilize the nonlinear system(1), where $\theta = \left[\theta_1,\theta_2,\cdots,\theta_i,\cdots,\theta_p\right]^T \left(i \in [1,2,\cdots,p]\right)$ is a parameter vector of p dimension which determines a unique equilibrium vector.

Assumption 2: There is a smooth function $x_e : R^p \to R^n$ such that:

$$f\left(x,\alpha\left(x,\theta\right)\right) = 0 \leftrightarrow x = x_e\left(\theta\right)$$

Assumption 3: The static performance map at the equilibrium point $x_e\left(\theta\right)$ from θ to y represented by:

$$y = F\left(x_e\left(\theta\right)\right) = F\left(\theta\right) \tag{3}$$

is smooth and has a unique global maximum or minimum vector $\theta^* \in R^p, \theta^* = \left[\theta_1^*, \theta_2^*, \cdots, \theta_p^*\right]^T$ such that:

$$\frac{\partial F\left(\theta^*\right)}{\partial \theta_i} = 0, \left(i = 1, 2, \cdots, p\right)$$

$$\text{and } \frac{\partial^2 F\left(\theta^*\right)}{\partial \theta_i^2} < 0 \text{ or } \frac{\partial^2 F\left(\theta^*\right)}{\partial \theta_i^2} > 0$$

Differentiating (3) with respect to time yields the relation between $\dot{\theta}$ and $\dot{y}\left(t\right)$.

$$J\left(\theta\left(t\right)\right)\dot{\theta}\left(t\right) = \dot{y}\left(t\right) \tag{4}$$

where $J\left(\theta(t)\right) = \left[\frac{\partial F(\theta)}{\partial \theta_1}, \frac{\partial F(\theta)}{\partial \theta_2}, \cdots, \frac{\partial F(\theta)}{\partial \theta_p}\right]^T$, $\dot{\theta}(t) = \left[\dot{\theta}_1, \dot{\theta}_2, \cdots, \dot{\theta}_p\right]^T$ and $\left|J\left(\theta\right)\right| = \left[\left|\frac{\partial F(\theta)}{\partial \theta_1}\right|, \left|\frac{\partial F(\theta)}{\partial \theta_2}\right|, \cdots, \left|\frac{\partial F(\theta)}{\partial \theta_p}\right|\right]^T$.

By Assumption 3, if the system converges to a global extreme vector θ^*, at the same time $\left|J\left(\theta\right)\right|$ will also converge to zero. A CARNN is applied to ESA to minimize $\left|J\left(\theta\right)\right|$ in finite time. Certainly the system is subjected to (4).

Then, the optimization problem can be written as follows.

Minimize: $c^T \xi$

Subject to: $A\xi = b$ (5)

where, $\xi = \left[J\left(\theta\right) \quad \left|J\left(\theta\right)\right| \quad \dot{\theta}\left(t\right)\right]^T$, $c = \left[0_{1\times p} \quad 1_{1\times p} \quad 0_{1\times p}\right]^T$, $b = \left[0 \quad \dot{y}(t) \quad \dot{y}(t)\right]^T$,

$$A = \begin{bmatrix} 1_{1\times p} & -sign\left(J^T\left(\theta\right)\right) & 0_{1\times p} \\ \dot{\theta}^T\left(t\right) & 0_{1\times p} & 0_{1\times p} \\ 0_{1\times p} & 0_{1\times p} & J^T\left(\theta\right) \end{bmatrix}, \text{ and } \operatorname{sign}\left(x\right) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}.$$

By the dual theory, the dual program corresponding to the program (5) is

Maximize: $b^T z$

Subject to: $A^T z = c$ (6)

where, $z^T = \left[z_1 \quad z_2 \quad z_3\right]_{1\times 3}$.

Therefore, an extremum seeking problem is converted into the programs defined in (5) and (6).

3 Chaotic Annealing Recurrent Neural Network Descriptions

3.1 Energy Function

In view of the primal and dual programs (5) and (6), define the following energy function:

$$E(\xi,z)=\frac{1}{2}\left(c^T\xi-b^Tz\right)^2+\frac{1}{2}\left\|T(t)\left(A\xi-b\right)\right\|^2+\frac{1}{2}\left\|T(t)\left(A^Tz-c\right)\right\|^2 \tag{7}$$

Clearly, the energy function (7) is convex and continuously differentiable. The first term in (7) is the squared difference between the objective functions of the programs (5) and (6), respectively. The second and the third terms are for the equality constraints of (5) and (6).

In order to gain the global extreme vector or approximate global extreme vector of the system, RNN is combined with chaotic annealing to construct a chaotic annealing parameter matrix $T(t)=diag\left(\eta_1e^{-\beta_1t},\eta_2e^{-\beta_2t},\eta_3e^{-\beta_3t}\right)_{3\times3}$, which is described as follow

$$\eta_i(t)=\frac{1}{1+e^{(-\sigma_i(t)/\varepsilon_i)}}(b_i-a_i)+a_i \tag{8}$$

$$\sigma_i'(t+1)=k\sigma_i(t)+a\left(\sum_{j=1,j\neq i}^{3}\omega_{ij}\eta_j(t)+I_i\right) \tag{9}$$

$$\sigma_i(t+1)=\begin{cases}\sigma_i'(t+1)+\gamma_i(t)\tau_i(t) & rand<P_i(t)\\ \sigma_i'(t+1) & otherwise\end{cases} \tag{10}$$

$$\gamma_i(t+1)=(1-\kappa)\gamma_i(t) \tag{11}$$

$$P_i(t+1)=\begin{cases}P_i(t)-\delta & P_i(t)>0\\ 0 & otherwise\end{cases} \tag{12}$$

$$\tau_i(t+1)=\rho(t)\tau_i(t)(1-\tau_i(t)) \tag{13}$$

where η_i denotes the output of the i th neuron. σ_i denotes the interior state of the i th neuron. σ_i' is a transitional variable. ε_i is a constant. a_i,b_i denote the above bound and the low bound of η_i respectively. ω_{ij} denotes the weight from the j th neuron to the i th neuron. I_i is the threshold value of the i th neuron. a is a proportion parameter. $k\left(0<k<1\right)$ is the decaying factor of the neuron. $\gamma_i(t)\left(\gamma_i(t)>0\right)$ is the chaotic coefficient. $\kappa\left(0<\kappa<1\right)$ is the decaying factor of $\gamma_i(t)$. $P_i(t)$ denotes the accepting probability of the chaotic coefficient $\gamma_i(t)$. $rand\left(0<rand<1\right)$ is a random number. $\delta(0<\delta<1)$ is a constant. $\tau_i(t)$ denotes the chaos noise of the i th neuron produced by iterated functions of Logistic map. $\tau_i(t)$ will gradually converge to the equilibrium points τ_i^*. If $\rho=4.0$ comes into existence, the logistic map of (13) will be full of the area$[0,1]$. The iterated function of Logistic map (13) is an invitation to the chaos phenomenon of CARNN. Because of the introduction of chaotic annealing parameter matrix $T(t)$, the searching process was divided into two phases: the coarse search based on chaos and the elaborate search based on RNN. Finally, CARNN will drive the system to stabilize at the global extreme point.

3.2 CARNN Architecture

With the energy function defined in (7), the dynamics for CARNN solving (5) and (6) can be defined by the negative gradient of the energy function as follows:

$$\frac{dv}{dt} = -\mu\nabla E(v) \tag{14}$$

where, $v=(\xi,z)^T$,$\nabla E(v)$ is the gradient of the energy function $E(v)$ defined in (7), and μ is a positive scalar constant, which is used to scale the convergence rate of the recurrent neural network.

The dynamical equation (14) can be expressed as:

$$\begin{aligned}
&\frac{du_1}{dt} = -\mu\left(cc^T + A^T T^T(t) T(t) A\right)v_1 + \mu b^T v_2 + \mu A^T T^T(t) T(t) b \\
&\frac{du_2}{dt} = \mu bc^T v_1 - \mu\left(bb^T + T^T(t) T(t) AA^T\right)v_2 + \mu T^T(t) T(t) Ac \\
&v_1 = u_1 \\
&v_2 = u_2
\end{aligned} \tag{15}$$

where, (u_1,u_2) is a column vector of instantaneous net inputs to neurons, (v_1,v_2) is a column output vector and equals to (ξ,z).

CARNN is described as the equation (15), which is determined by the number of decision variables such as (ξ,z). The lateral connection weight matrix is defined as $\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \begin{bmatrix} -\mu\left(cc^T + A^T T^T(t) T(t) A\right) & \mu b^T \\ \mu bc^T & -\mu\left(bb^T + T^T(t) T(t) AA^T\right) \end{bmatrix}$, the biasing threshold vector of the neurons is defined as $\begin{bmatrix} \vartheta_1 \\ \vartheta_2 \end{bmatrix} = \begin{bmatrix} \mu A^T T^T(t) T(t) b \\ \mu T^T(t) T(t) Ac \end{bmatrix}$. By adjusting μ and$T(t)$, the weight matrix and the biasing threshold vector can be adjusted.

4 Convergence Analysis

We analyze the stability of the proposed CARNN controller in the section.

Lemma 1[6]: Suppose that $f: D \subset R^n \rightarrow R$ is differentiable on a convex set $D_0 \subset D$. Then f is convex on D_0 if and only if

$$(z-y)^T \nabla f(y) \leq f(z) - f(y), \ \forall y, z \in D_0 \tag{16}$$

where $\nabla f(y)$ is the gradient of $f(y)$.

Lemma 2: The optimal solution to the programs (5) and (6) are ξ^* and z^*, respectively, if and only if $E(v^*) = 0$ and

$$(v^* - v)^T \nabla E(v,t) \leq -E(v,t) \tag{17}$$

where $v^* = (\xi^{*T}, z^{*T})^T$ and $v = (\xi^T, z^T)^T$.

Proof: Form the definition of the energy function (7), it can easily find that $E(v^*) = 0$ if and only if v^* is the optimal solution of (5) and (6). Since for all v, the energy function $E(u,t) \geq 0$ is continuously differentiable and convex. Therefore, we have the conclusion of the Lemma 2 from Lemma 1.

Theorem: CARNN defined in (15) is globally stable and converges to the optimal solutions of the program (5) and (6).

Proof: Without loss of generality, let $\mu = 1$. Consider the following Lyapunov function:

$$V(v) = \frac{1}{2}(v^* - v)^T (v^* - v) \tag{18}$$

Where $v^* = (\xi^{*T}, z^{*T})^T$, and ξ^*, z^* are the optimal solutions to the programs (5) and (6), respectively. By Lemma 2 and the equation (14), we have

$$\frac{dV}{dt} = \frac{dV}{dv}\left(\frac{dv}{dt}\right) = -(v^* - v)^T \frac{dv}{dt} = (v^* - v)^T \nabla E(v) \leq -E(v) \leq 0 \tag{19}$$

Since $E(v) \geq 0$, according to the Lyapunov's theorem, CARNN defined in (15) is Lyapunov stable. From Lemma 2, $E(v^*) = 0$ if and only if $\nabla E(v^*) = 0$. Hence v^* makes $\dot{v} = 0$ and $\dot{V} = 0$, and therefore CARNN defined in (15) converges to its equilibrium points, and then $|J(\theta)|$ converges to its minimum point. So $\theta = \theta^*$, the output y of the system (1) equals to the optimal solution $y^* = F(\theta^*)$.

Since $E(v,t)$ is continuously differentiable and convex for all v, the local minimum is equivalent to the global minimum. CARNN defined in (15) is thus globally stable and converges to the optimal solutions of the programs (5) and (6). The proof is completed.

5 Simulation Results

5.1 A Simplified Tight Formation Flight Model Simulation

Consider a simplified tight formation flight model consisting of two Unmanned Aerial Vehicles [7].

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -20 & -9 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -35 & -15 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{20}$$

with a cost function given by

$$y(t) = -10(x_1(t)+0)^2 - 5(x_3(t)+9)^2 + 590 \tag{21}$$

where x_1 is the vertical separation of two Unmanned Aerial Vehicles, x_2 is the differential of x_1, x_3 is the lateral separation of two Unmanned Aerial Vehicles, x_4 is the differential of x_3 and y is the upwash force acting on the wingman. It is clear that the global maximum point is $x_1^* = 0$ and $x_3^* = -9$, where the cost function $y(t)$ reaches its maximum $y^* = 590$.

A control law based on sliding mode theory is given by:

$$\begin{cases} u_1 = 20x_1 + (9-s_1)x_2 - k_1 sign(\sigma_1 - s_1\theta_1), \sigma_1 = s_1x_1 + x_2 \\ u_2 = 35x_3 + (15-s_2)x_4 - k_2 sign(\sigma_2 - s_2\theta_2), \sigma_2 = s_2x_3 + x_4 \end{cases} \tag{22}$$

where σ_1, σ_2 are two sliding mode surfaces, s_1, k_1, s_2, k_2 are positive scalar constants, θ_1, θ_2 are two extremum seeking parameters, which a CARNN is used to seek at the same time.

Remark: The control law is given in (22), which is based on sliding mode theory. We choose $sign(\sigma_i - s_i\theta_i), (i = 1,2)$ so that x_1 and x_3 entirely traces θ_1 and θ_2 in the sliding mode surfaces respectively, and the system will be stable at θ_1^* and θ_2^* finally.

The initial conditions of the system (20) are given as $x_1(0) = -2$, $x_2(0) = 0$, $x_3(0) = -4$, $x_4(0) = 0$, $\theta_1(0) = -2$, $\theta_2(0) = -4$. Applying CARNN to system (20), the parameters are given as: $\beta_1 = 0$, $\beta_2 = 0$, $\beta_3 = -0.01$, $s_1 = 2.15$, $s_2 = 3.35$, $k_1 = k_2 = 1$, $\mu = 0.235$, $a = 0.15$, $k = 0.9$, $\sigma = [0.15, 0.15, 0.15]$, $a_1 = a_2 = a_3 = 0$, $b_1 = b_2 = b_3 = 1$, $\omega_{12} = \omega_{21} = 0.2$, $\omega_{23} = \omega_{32} = 0.2$, $\omega_{31} = \omega_{13} = 0.1$, $\omega_{11} = \omega_{22} = \omega_{33} = 0$, $I_1 = I_2 = I_3 = 0.01$, $\varepsilon = [0.1, 0.1, 0.1]$, $P(0) = [1,1,1]$, $\delta = 0.01$, $\kappa = 0.01$, $\gamma(0) = [0.1, 0.1, 0.1]$, $\rho(0) = 4.0$, $\tau(0) = [0.875, 0.875, 0.513]$. Certainly, μ is a main factor of scaling the convergence rate of CARNN, if it is too big, the error of the output will be introduced, on the contrary, if it is too small, the convergence rate of the system will be slow. The values of β_i should not be too big, otherwise the system will be unstable. In conclusion, the values of those parameters should be verified by the system simulation. The simulation results are shown from figure 1 to figure 3.

In those figures, solid lines are the results applying CARNN to ESA; dash lines are the results applying ESA with sliding mode [8]. Comparing those simulation results, we know the dynamic performance of the method proposed in the paper is better than that of ESA with sliding mode. The "chatter" of the CARNN's output doesn't exist in figure 2 and 3, which is very harmful in practice. Moreover the convergence rate of ESA with CARNN can be scaled by adjusting the chaotic annealing matrix.

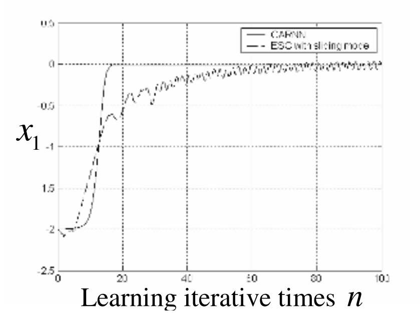


Fig. 1. The result of the state x_1

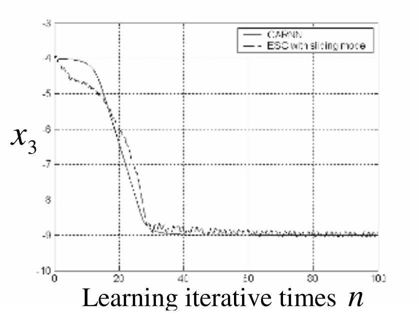


Fig. 2. The result of the state x_3

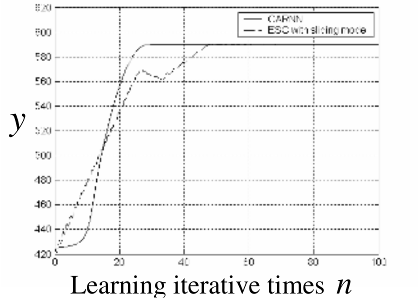


Fig. 3. The result of the output y

5.2 A Testing Function Simulation

In order to exhibit the capability of global searching of the proposed CARNN, the typical testing function (23) is defined as the cost function of system (20).

$$max\, y(t) = -(x_1 - 0.7)^2\left((x_3 + 0.6)^2 + 0.1\right) - (x_3 - 0.5)^2\left((x_1 + 0.4)^2 + 0.15\right) \tag{23}$$

The above function have a global maximum point $(0.7, 0.5)$ and three local maximum, which are $(0.6, 0.4)$, $(0.6, 0.5)$ and $(0.7, 0.4)$ respectively. The initial conditions of the system (23) are given as $x_1(0) = 0.2$, $x_2(0) = 0$, $x_3(0) = 0.7$, $x_4(0) = 0$, $\theta_1(0) = 0.2$, $\theta_2(0) = 0.7$ .CARNN is applied to solve the problem. By choosing appropriately those parameters of the system, the simulation results are shown from figure 4 to figure 6.

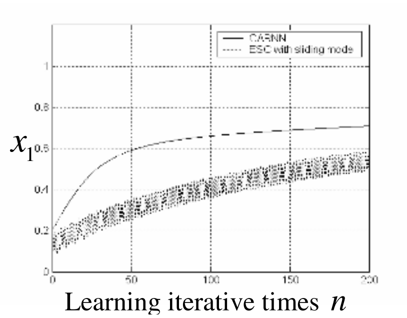


Fig. 4. The result of the state x_1

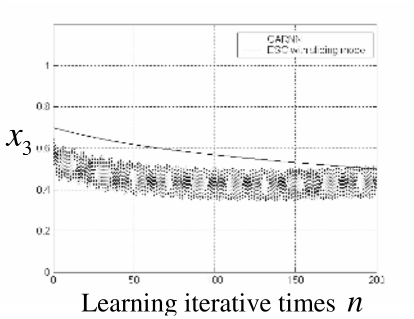


Fig. 5. The result of the state x_3

In those figures, solid lines are the results applying CARNN to ESA; dash lines are the results applying ESA with sliding mode [8]. Comparing those simulation results, we know that CARNN drives the system to the global extreme point by finite iterative times, but ESA with sliding mode traps the system into a local extreme point and results in the serious "chatter". Hence the capability of global searching of CARNN is validated.

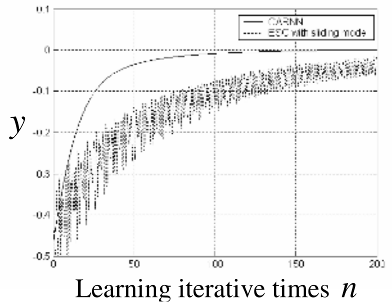


Fig. 6. The result of the output y

6 Conclusion

The method of combining CARNN with ESA greatly improves the dynamic performance and the global searching capability of the system. Two phases of the coarse search based on chaos and the elaborate search based on RNN ensure that the system could fully carry out the chaos searching and find the global extremum point and accordingly converge to that point. At the same time, the disappearance of the "chatter" of the system output and the switching of the control law are beneficial to engineering applications.

References

1. Blackman, B. F.: Extremum-seeking Regulators. An Exposition of Adaptive Control, New York: Macmillan (1962) 36-50.
2. Drakunov, S., Ozguner, U., Dix, P., and Ashrafi, B.: ABS Control Using Optimum Search via Sliding Mode., IEEE Transactions on Control Systems Technology, Vol. 3, No. 1 (1995) 79-85.
3. Krstic, M.: Toward Faster Adaptation in Extremum Seeking Control. Proc. of the 1999 IEEE Conference on Decision and Control, Phoenix. AZ (1999) 4766-4771.
4. Tan, Y., Wang, B.Y., He, Z.Y.: Neural Networks with Transient Chaos and Time-variant gain and Its Application to Optimization Computations. ACTA ELECTRONICA SINICA, Vol. 26, No. 7 (1998) 123-127.
5. Wang, L., Zheng, D.Z.: A Kind of Chaotic Neural Network Optimization Algorithm Based on Annealing Strategy. Control Theory and Applications, Vol. 17, No. 1 (2000) 139-142.
6. Tang, W.S. and Wang, J.: A Recurrent Neural Network for Minimum Infinity-Norm Kinematic Control of Redundant Manipulators with an Improved Problem Formulation and Reduced Architecture Complexity. IEEE Transactions on systems, Man and Cybernetics, Vol. 31, No. 1 (2001) 98-105.
7. Zuo, B. and Hu, Y.A.: Optimizing UAV Close Formation Flight via Extremum Seeking. WCICA2004, Vol. 4. 3302-3305.
8. Pan, Y., Ozguner, U., and Acarman, T.: Stability and Performance Improvement of Extremum Seeking Control with Sliding Mode. Control. Vol. 76 (2003) 968-985.

Improved Transiently Chaotic Neural Network and Its Application to Optimization

Yao-qun Xu[1,2], Ming Sun[1], and Meng-shu Guo[2]

[1] Institute of System Engineering, Harbin University of Commerce, 150028, Harbin, China
Xuyq@hrbcu.edu.cn, Snogisun@tom.com
[2] Center for Control Theory and Guidance Technology, Harbin Institute of Technology, 150001
Harbin, China
Xyqcx02@yahoo.com.cn

Abstract. A wavelet function was introduced into the activation function of the transiently chaotic neural network in order to solve combinational optimization problems more efficiently. The dynamic behaviors of chaotic signal neural units were analyzed and the time evolution figures of the maximal Lyapunov exponents and chaotic dynamic behavior were given. The improved transiently chaotic neural network has the ability to stay in chaotic states longer because the wavelet function is non-monotonous and is a kind of basic function. The simulation results prove that the improved transiently chaotic neural network is superior to the original in solving 10-city traveling salesman problem (TSP).

1 Introduction

Neural network is a very complicate nonlinear system, and it contains all kinds of dynamic behaviors. Some chaotic behaviors have been observed in human brains and animals' neural systems, so it would improve the intelligent ability in neural network and artificial neural network would have much more use in application if chaotic dynamics mechanism is introduced into artificial neural network. Chaotic neural networks have been proved to be powerful tools for escaping from local minimum. Chaotic neural networks with chaotic dynamics have much rich and far-from equilibrium dynamics with various coexisting attractors, not only of fixed and periodic points but also of strange attractors. By far, many transiently neural network models have been presented [1~3]. In this paper, Morlet wavelet function was introduced into the activation function of the transiently chaotic neural network, and the time evolution figures of the maximal Lyapunov exponents and chaotic dynamic behavior of chaotic single neural unit were given. The improved transiently chaotic neural network has the ability to stay in chaotic states longer because the wavelet function is non-monotonous and is a kind of basic function. The simulation results prove that the improved transiently chaotic neural network is superior to the original in solving 10-city traveling salesman problem (TSP).

For any function $f(x) \in L_2(R)$ and any wavelet Ψ as a kind of basic function, the known formula can be described as follows.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1032 – 1041, 2006.

$$f(x) = \sum_{j,k=-\infty}^{\infty} c_{j,k} \Psi_{j,k}(x) \tag{1}$$

2 Chaotic Neural Network Models

In this section, two chaotic neural network models are given. The first is presented by Li-jiang Yang, the second is improved model by introducing Morlet wavelet function in activation function of Yang's.

2.1 Yang's Transiently Chaotic Neural Network [4]

Li-jiang Yang, Tian-Lun Chen and Wu-qun Huang's transiently chaotic neural network is described as follows:

$$x_i(t) = \frac{1}{1+e^{-y_i(t)/\varepsilon}} \tag{2}$$

$$y_i(t+1) = ky_i(t) + \alpha\left[\sum_{j=1}^{N} W_{ij} x_i(t) + I_i\right] + z(t)g[y_i(t) - y_i(t-1)] \tag{3}$$

$$g(x) = 5xe^{-5|x|} \tag{4}$$

$$z_i(t+1) = (1-\beta)z_i(t) \tag{5}$$

where $x_i(t)$ is output of neuron i ; $y_i(t)$ denotes internal state of neuron i ; W_{ij} describes connection weight from neuron j to neuron i , $W_{ij} = W_{ji}$; I_i is input bias of neuron i , α a positive scaling parameter for neural inputs, k damping factor of nerve membrane, 0≤ k ≤1, ε steepness parameter of the activation function, ε >0.

The chaotic neural network is different from the other chaotic neural network in the right of the equation (3), the self-feedback connection weight $g[y_i(t) - y_i(t-1)]$. It is just the self-feedback connection weight that makes the chaotic neural network embrace the rich chaotic dynamics.

The self-feedback connection weight should take the non-linear form. The form of non-linear function $g(x)$ is chosen under the following consideration [5]: it should not change the fixed points of the equation (3) but the stability of the fixed points may be changed. This demands that $g(0)$ =0.

2.2 Improved Transiently Chaotic Neural Network [6]

The improved transiently chaotic neural network is described as follows:

$$x_i(t) = \frac{1}{1+e^{-y_i(t)/\varepsilon_1}} + \frac{1}{3} Morlet(y_i(t)/\varepsilon_2) \tag{6}$$

$$y_i(t+1) = ky_i(t) + \alpha\left[\sum_{j=1}^{N} W_{ij} x_i(t) + I_i\right] + z(t)g[y_i(t) - y_i(t-1)] \tag{7}$$

$$g(x) = 5xe^{-5|x|} \tag{8}$$

$$Morlet(x) = e^{-x^2/2}\cos(5x) \tag{9}$$

$$z_i(t+1) = (1-\beta)z_i(t) \tag{10}$$

where $x_i(t)$, $y_i(t)$, W_{ij} , α , k , I_i are the same with the above. ε_1 and ε_2 are respectively the steepness parameters of Sigmoid function and Morlet wavelet function.

The improved transiently chaotic neural network has a non-monotonous activation function, which is composed of Sigmoid and Morlet wavelet. Several kinds of chaotic neural networks whose activation function is non-monotonous has been proved to be more powerful than Chen's chaotic neural network in solving optimization problems, especially in searching global minima of continuous function and traveling salesman

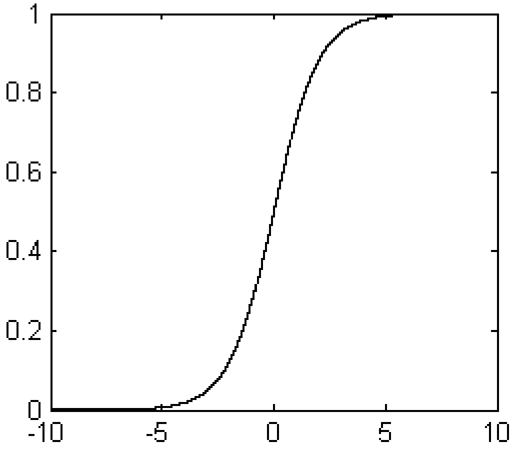


Fig. 1. The monotonous figure of sigmoid function

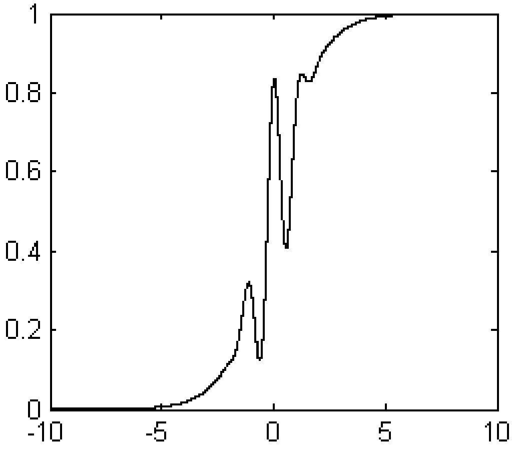


Fig. 2. The non-monotonous figure of the function composed of sigmoid and Morlet wavelet

problems [6-8]. The reference [9] has pointed out that the single neural unit can easily behave chaotic behavior if its activation function is non-monotonous. And the reference [10] has presented that the effective activation function may adopt kinds of different forms, and should show non-monotonous behavior. Not only is the activation function a non-monotonous function, but also the Morlet wavelet of the activation function is a kind of basic function. The figures of Sigmoid function and the function composed of Sigmoid and Morlet wavelet are respectively plotted as fig.1 and fig.2. Seen from the fig.2, the activation function of the improved model is non-monotonous.

3 The Dynamic Analyses of Chaotic Neural Networks

In this section, the chaotic dynamic behaviors of the chaotic neural units are analyzed, and the time evolution figures of the maximal Lyapunov exponents and chaotic dynamic behavior are given.

3.1 Yang's Chaotic Signal Neural Unit

The signal neural unit model can be described as follows:

$$x(t) = \frac{1}{1+e^{-y(t)/\varepsilon}} \tag{11}$$

$$y(t+1) = ky(t) + g[y(t) - y(t-1)] \tag{12}$$

$$g(x) = 5xe^{-5|x|} \tag{13}$$

$$z(t+1) = (1-\beta)z(t) \tag{14}$$

The parameters are set as follows:

k =0.5, ε =1/20 ,z=10, y(0)=0.283, $\beta = 0.008$.

The time evolution figures of Lyapunov exponents and chaotic dynamic behavior are shown as Fig.3, Fig.4:

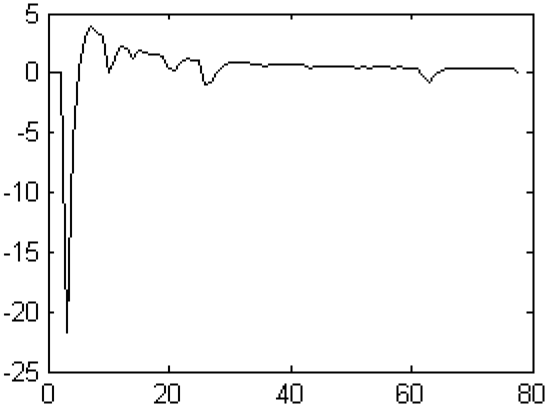


Fig. 3. The time evolution figure of Lyapunov exponents

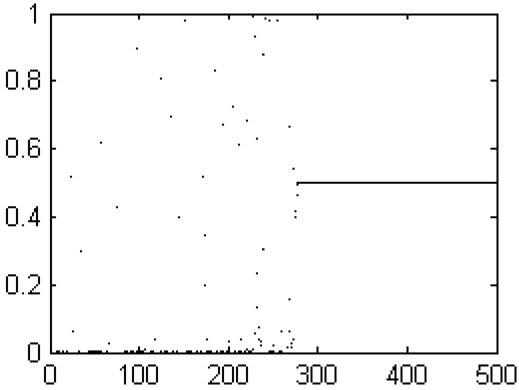


Fig. 4. The chaotic dynamic behavior of x

3.2 Improved Chaotic Signal Neural Unit

The signal neural unit model can be described as follows:

$$x(t)=\frac{1}{1+e^{-y(t)/\varepsilon}}+\frac{1}{3}Morlet(y(t)) \tag{15}$$

$$y(t+1)=ky(t)+g[y(t)-y(t-1)] \tag{16}$$

$$g(x)=5xe^{-5|x|} \tag{17}$$

$$Morlet(x)=e^{-x^2/2}\cos(5x) \tag{18}$$

$$z(t+1)=(1-\beta)z(t) \tag{19}$$

The parameters are set as follows:

k =0.5, ε_1 =1/20 , ε_2 =5/4, z=10, y(0)=0.283, $\beta = 0.008$.

The time evolution figures of Lyapunov exponents and chaotic dynamic behavior are shown as Fig.5, Fig.6.

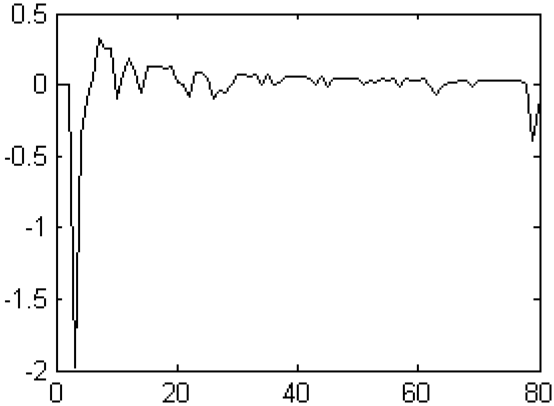


Fig. 5. The time evolution figure of Lyapunov exponents

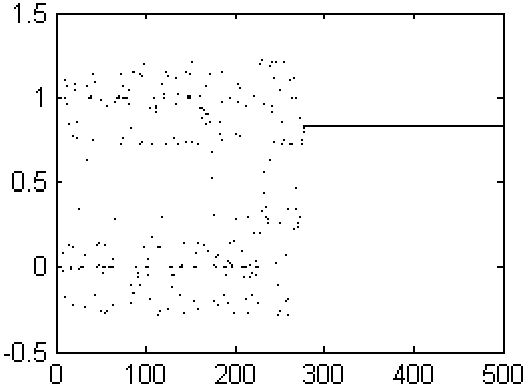


Fig. 6. The chaotic dynamic behavior of x

Seen from the Fig.4-Fig.6, the chaotic dynamic mechanism functions. But the equilibrium point's form of this chaotic search is different from the form of the reversed bifurcation. It seems to suddenly reach an equilibrium point after the chaotic search.

4 Application to Traveling Salesman Problem

The coordinates of 10-city is as follows:
(0.4, 0.4439),(0.2439, 0.1463),(0.1707, 0.2293),(0.2293, 0.716),(0.5171,0.9414), (0.8732, 0.6536), (0.6878, 0.5219), (0.8488, 0.3609),(0.6683, 0.2536),(0.6195, 0.2634). The shortest distance of the 10-city is 2.6776.

A solution of TSP with N cities is represented by N×N-permutation matrix, where each entry corresponds to output of a neuron in a network with N×N lattice structure. Assume v_{xi} to be the neuron output which represents city x in visiting order i . A computational energy function which is to minimize the total tour length while simultaneously satisfying all constrains takes the follow form [11]:

$$E=\frac{A}{2}(\sum_{x=1}^{N}(\sum_{i=1}^{N}v_{xi}-1)^2+\sum_{i=1}^{N}(\sum_{x=1}^{N}v_{xi}-1)^2)+\frac{B}{2}\sum_{x=1}^{N}\sum_{y=1}^{N}\sum_{i=1}^{N}d_{xy}v_{xi}v_{y,i+1} \tag{20}$$

Where $v_{x0}=v_{xN}$ and $v_{x,N+1}=v_{x1}$. A and B are the coupling parameters corresponding to the constrains and the cost function of the tour length, respectively. d_{xy} is the distance between city x and city y .

The parameters of the energy function are set as follows: A=2.5, B=1.

In this paper, the improved chaotic neural network and the original are compared by the different steepness parameter of Sigmoid function. So the rest parameters retain unchanged.

(1) The parameters of Yang's are set as follows:

α=0.5, k=1, ε=1/20,z(0)=0.08, $\beta=0.008$.

The parameters of the improved chaotic neural network are set as follows:

α=0.5, k=1, ε_1=1/20, ε_2= 5/4, z(0)=0.08, $\beta=0.008$.

200 different initial conditions of y_{ij} are generated randomly in the region [0, 1], as is shown in table 1. (VN= valid number; GN= global number; VP= valid percent; GP=global percent.)

Table 1. The 10-city simulation result as ε =1/20

Model	VN	GN	VP	GP	Model	VN	GN	VP	GP
Yang's	175	125	87.5%	62.5%	Improved Chaotic Neural network	187	182	93.5%	91%
	179	137	89.5%	68.5%		190	185	95%	92.5%
	173	134	86.5%	67.5%		189	184	94.5%	92%
	179	134	89.5%	67.5%		191	184	95.5%	92%
	179	119	89.5%	59.5%		192	184	96%	92%
	181	122	90.5%	61%		187	174	93.5%	87%
	187	131	93.5%	65.5%		190	181	95%	90.5%
	185	135	92.5%	67.5%		185	179	92.5%	89.5%
	185	122	92.5%	61%		189	187	94.5%	93.5%
	180	123	90%	61.5%		192	186	96%	93%

(2) The parameters of Yang's are set as follows:

α=0.5, k =1, ε =1/10,z(0)=0.08, $\beta = 0.008$.

The parameters of the improved chaotic neural network are set as follows:

α=0.5, k =1, ε_1 =1/10, ε_2 = 5/4, z(0)=0.08, $\beta = 0.008$.

200 different initial conditions of y_{ij} are generated randomly in the region [0, 1], as is shown in table 2. (VN= valid number; GN= global number; VP= valid percent; GP=global percent.)

Table 2. The 10-city simulation result as ε =1/10

Model	VN	GN	VP	GP	Model	VN	GN	VP	GP
Yang's	187	151	83.5%	75.5%	Improved Chaotic Neural network	193	177	96.5%	88.5%
	188	150	94%	75%		188	174	94%	87%
	188	158	94%	79%		194	188	97%	94%
	182	149	91%	74.5%		190	174	95%	87%
	187	152	93.5%	76%		188	179	94%	89.5%
	180	148	90%	74%		193	179	96.5%	89.5%
	188	152	94%	76%		189	174	94.5%	87%
	184	150	92%	75%		192	180	96%	90%
	180	148	90%	74%		191	181	95.5%	90.5%
	182	155	91%	75.5%		193	188	96.5%	94%

(3) The parameters of Yang's are set as follows:

α=0.5, k =1, ε =1/5,z(0)=0.08, $\beta = 0.008$.

The parameters of the improved chaotic neural network are set as follows:

α=0.5, k =1, ε_1=1/5, ε_2= 5/4, z(0)=0.08, β = 0.008 .

200 different initial conditions of y_{ij} are generated randomly in the region [0, 1], as is shown in table 3. (VN= valid number; GN= global number; VP= valid percent; GP=global percent.)

Table 3. The10-city simulation result as ε =1/5

Model	VN	GN	VP	GP	Model	VN	GN	VP	GP
Yang's	181	151	90.5%	75.5%	Improved Chaotic Neural network	187	169	93.5%	88.5%
	182	152	91%	76%		189	169	94.5%	88.5%
	189	157	94.5%	78.5%		191	179	95.5%	89.5%
	179	147	89.5%	73.5%		180	169	90%	84.5%
	181	162	90.5%	81%		186	179	93%	89.5%
	179	143	89.5%	71.5%		192	178	96%	89%
	180	137	90%	68.5%		184	170	92%	85%
	179	154	89.5%	77%		185	170	92.5%	85%
	185	156	92.5%	78%		186	172	93%	86%
	184	158	92%	79%		184	168	92%	84%

Seen from the above tables, the conclusion can be drawn that the improved transiently chaotic neural network has the stronger ability to solve TSP when the parameters of the two networks are in the above same level.

The time evolution figures of Energy function are given as follows:

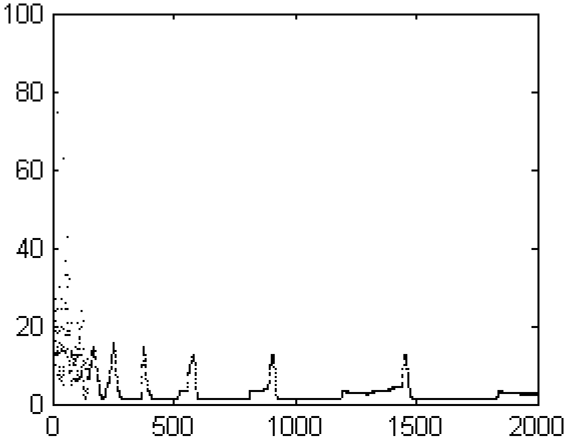


Fig. 7. Time evolution of energy in Yang's

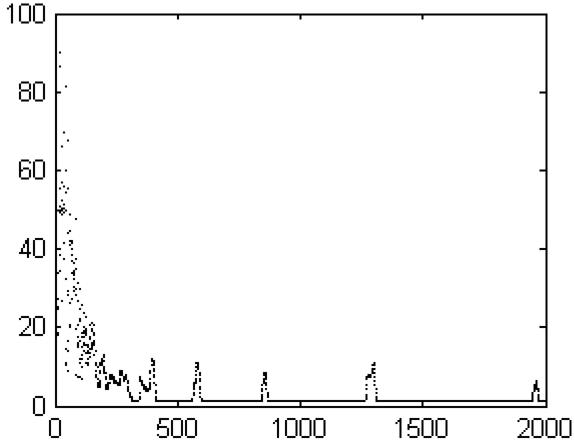


Fig. 8. Time evolution of energy in the improved

However, different networks may reach their best performance at different parameters. How do these networks make comparison with the same parameters? In this paper, the question needs to solve. In this paper, the test only shows that under these parameters the improved model is superior to the original model.

5 Conclusions

The improved transiently chaotic neural network is superior to the original network under the same parameters, and this owe to Morlet wavelet function which is non-monotonous. However, sometimes the improved transiently chaotic neural network is not superior to the original network, and even inferior to the original network. So, the improved transiently chaotic neural network should be made further research.

Acknowledgement

This work is supported by the Youth Science Foundation of Harbin 2005AFQXJ040.

References

1. Chen L., Aihara K.:Chaotic Simulated Annealing by a Neural Network Model with Transient Chaos. Vol. 8. Neural Networks. (1995)915-930
2. Yamada T, Aihara K, Kotani M.: Chaotic Neural Networks and The Travelling Salesman Problem. Proceedings of 1993 International Joint Conference on Neural Networks, 1993. 1549-1552p
3. Aihara K, Takabe T, Toyada M.: Chaotic Neural Networks. Phys. Letters A, 1990,144(6/7): 333 - 340.

4. Yang Lijiang, Chen Tianlun, Huang Wuqun.: Application of Transiently Chaotic Dynamics in Neural Computing. Acta Scientiarum Naturalium Universitatis Nankaiensis. 1999, 99-103
5. Zhou Chang-song, Chen Tian-lun, Huang Wu-qun.: Chaotic neural network with nonlinear self-feedback and its application in optimization
6. Y.-q. xu, M. Sun, G.-r. Duan.: Wavelet Chaotic Neural Networks and Their Application to Optimization problems.ISNN2006, LNCS, Vol. 3791. Springer (2006) 379-384
7. Y.-q. Xu and M. Sun.: Gauss-Morlet-Sigmoid Chaotic Neural Networks. ICIC 2006,LNCS, vol.4113,Springer-Verlag Berlin Heidelberg (2006) 115-125
8. Y.-q. Xu, M. Sun, and J.-h. Shen.: Gauss Chaotic Neural Networks. PRICAI 2006, LNAI , vol.4099,Springer-Verlag Berlin Heidelberg (2006) 319-328
9. A Potapove, M Kali.: Robust chaos in neural networks. Physics Letters A, vol.277,no.6, (2000)310-322
10. Shuai J W, Chen Z X, Liu R T, et al.: Self-evolution Neural Model. Physics Letters A, vol.221,no.5, (1996)311-316
11. S.-y. Sun, J.-l. Zheng.: A Kind of Improved Algorithm and Theory Testify of Solving TSP in Hopfield Neural Network. Vol.1. Journal of Electron. (1995)73-78

Quantum-Behaved Particle Swarm Optimization for Integer Programming

Jing Liu, Jun Sun, and Wenbo Xu

Center of Intelligent and High Performance Computing,
School of Information Technology, Southern Yangtze University
No. 1800, Lihudadao Road, Wuxi,
214122 Jiangsu, China
{liujing_novem, sunjun_wx, xwb_sytu}@hotmail.com

Abstract. Based on our previously proposed Quantum-behaved Particle Swarm Optimization (QPSO), this paper discusses the applicability of QPSO to integer programming. QPSO is a global convergent search method, while the original Particle Swarm (PSO) cannot be guaranteed to find out the optima solution of the problem at hand. The application of QPSO to integer programming is the first attempt of the new algorithm to discrete optimization problem. After introduction of PSO and detailed description of QPSO, we propose a method of using QPSO to solve integer programming. Some benchmark problems are employed to test QPSO as well as PSO for performance comparison. The experiment results show the superiority of QPSO to PSO on the problems.

1 Introduction

An Integer programming problem is an optimization problem in which some or all of variables are restricted to take on only integer values. Thus the general form of a mathematical Integer Programming model can be stated as:

$$\begin{aligned} &\min_{x} \ f(x) \\ &s.t. \quad g(x) \le b, \quad x \in X \end{aligned} \tag{1}$$

where

$$X = \{x \in \Re^n : x_i \in Z^n, \forall i \in M\}, \ M \subseteq [1..n]$$

This type of model is called a mixed-integer linear programming model, or simply a mixed-integer program (MIP). If $M = [1...n]$, we have a pure integer linear programming model, or integer program (IP). Here we will consider only the simple and representative minimization IP case, though maximization IP problems are very common in the literature, since a maximization problem can be easily turned to a minimization problem. For simplicity, in this paper it will be assumed that all of the variables are restricted to be integer valued without any constraints.

Evolutionary and Swarm Intelligence algorithms are stochastic optimization methods that involve algorithmic mechanisms similar to natural evolution and social behavior respectively. They can cope with problems that involve discontinuous

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1042 – 1050, 2006.

objective functions and disjoint search spaces [7][8]. Early approaches in the direction of Evolutionary Algorithms for Integer Programming are reported in [9][10]. The performance of PSO method on Integer Programming problems was investigated in [6] and the results show that the solution to truncate the real value to integers seems not to affect significantly the performance of the method.

In this paper, the practicability of QPSO to integer programming is explored. For QPSO is global convergent, it can be expected to outperform PSO in this field. To test the algorithm, numerical experiment is implemented. The paper is organized as follows. In Section 2 we describe the concepts of QPSO. Section 3 presents the numerical results of both QPSO and PSO on several benchmark problems. The paper is concluded in Section 4.

2 Quantum-Behaved Particle Swarm Optimization

In this section, the concept of Quantum-behaved Particle Swarm Optimization is described following the introduction of the original Particle Swarm Optimization.

2.1 Particle Swarm Optimization

The PSO algorithm is population based stochastic optimization technique proposed by Kennedy and Eberhart in 1995[1]. The motivation for the development of this method was based on the simulation of simplified animal social behaviors such as fish schooling, bird flocking, etc.

In the original PSO model, each individual is treated as volumeless and defined as a potential solution to a problem in D-dimensional space, with the position and velocity of particle i represented as $X_i = (x_{i1}, x_{i2}, ... x_{iD})$ and $V_i = (v_{i1}, v_{i2}, ..., v_{iD})$, respectively. Each particle maintains a memeory of its previous best positon $P_i = (p_{i1}, p_{i2}, ..., p_{iD})$ and p_{gd}, designated g, represents the position with best fitness in the local neighborhood. The particle will move according to the following equation:

$$\begin{aligned} v_{id} &= v_{id} + \varphi_1 * rand(\ ) * (p_{id} - x_{id}) + \varphi_2 * rand(\ ) * (p_{gd} - x_{id}) \\ x_{id} &= x_{id} + v_{id} \end{aligned} \tag{2}$$

where φ_1 and φ_2 determine the relative influence of the social p_g and cognition p_i components, which are the embodiment of the spirit of cooperation and competition in this algorithm.

Since the introduction of PSO method in 1995, considerable work has been done in the aspect of improving its convergence, diversity and precision etc. Generally, in population-based search optimization methods, proper control of global exploration and local exploration is crucial in finding the optimum solution effectively. In[2] Eberhart and Shi show that PSO searches wide areas effectively, but tends to lack search precision. So they proposed the solution to introduce ω, a linearly varying inertia weight, that dynamically adjusted the velocity over time, gradually focusing PSO into a local search:

$$v_{id} = \omega * v_{id} + \varphi_1 * rand(\ ) * \left(p_{id} - x_{id}\right) + \varphi_2 * rand(\ ) * \left(p_{gd} - x_{id}\right) \tag{3}$$

The improved PSO is called Standard PSO algorithm(in this paper PSO-w denoted).

Then Maurice Clerc introduced a constriction factor[3] , K ,that improved PSO's ability to prevent the particles from exploding outside the desirable range of the search space and induce convergence. The coefficient K is calculated as:

$$K = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, \text{ where } \varphi = \varphi_1 + \varphi_2, \varphi > 4 \tag{4}$$

and the PSO is then

$$v_{id} = K * (v_{id} + \varphi_1 * rand(\ ) * (p_{id} - x_{id}) + \varphi_2 * rand(\ ) * (p_{gd} - x_{id})) \tag{5}$$

2.2 Quantum-Behaved Particle Swarm Optimization

Even though many improvements on PSO methods were emerged, some questions around traditional PSO still exist. In traditional PSO system, a linear system, a determined trajectory and the bound state is to guarantee collectiveness of the particle swarm to converge the optimal solution. However, in such ways, the intelligence of a complex social organism is to some extend decreased. Naturally, Quantum theory, following the Principle of State Superposition and Uncertainty, was introduced into PSO and the Quantum-behaved PSO algorithm was proposed by Jun Sun et al[4].

Keeping to the philosophy of PSO, a Delta potential well model of PSO in quantum world is presented, which can depict the probability of the particle's appearing in position x from probability density function $|\psi(x,t)|^2$, not limited to determined trajectory, with the center on point p(pbest). The wave function of the particle is:

$$\psi(x) = \frac{1}{\sqrt{L}} \exp\left(-\|p - x\| / L\right) \tag{6}$$

And the probability density function is

$$Q(x) = |\psi(x)|^2 = \frac{1}{L} \exp\left(-2\|p - x\| / L\right) \tag{7}$$

The parameter $L(t+1) = 2 * \alpha * |p - x(t)|$ depending on energy intension of the potential well specifies the search scope of a particle. From the expression of L, we can see that it is so unwise to deploy the individual's center pbest to the swarm that unstable and uneven convergence speed of an individual particle will result premature of the algorithm when population size is small. Then a conception of Mean Best Position (mbest) is introduced as the center-of-gravity position of all the particles [5]. That is

$$mbest = \sum_{i=1}^{M} p_i \Big/ M = \left(\sum_{i=1}^{M} p_{i1} \Big/ M, \sum_{i=1}^{M} p_{i2} \Big/ M, ..., \sum_{i=1}^{M} p_{id} \Big/ M\right) \tag{8}$$

here M is the population size and p_i is the pbest of particle i. Thus the value of L is given by $L(t+1) = 2 * \beta * |mbest - x(t)|$. We can see the only parameter in this algorithm is β, called Creativity Coefficient, working on individual particle's convergence speed and performance of the algorithm.

Through the Monte Carlo stochastic simulation method, derived from probability density function, the position of a particle that is vital to evaluate the fitness of a particle can be given by $x(t) = p \pm \frac{L}{2} \ln\left(1/u\right)$. Replacing parameter L, the iterative equation of Quantum-behaved PSO (denoted QPSO- β) is:

$$x(t+1) = p \pm \beta * |mbest - x(t)| * \ln\left(1/u\right) \tag{9}$$

3 Experiments

3.1 Experiment Setting and Benchmark Problems

The method of Integer Programming by PSO and QPSO algorithm is to truncate each particle of the swarm to the closest integer, after evolution according to Eq(2) and Eq(9). In our experiments, each algorithm was tested with all of the numerical test

Table 1. Benchmark problems

F	Mathematical Representation	Solution	Solution
F1	$F_1(x) = \|x\|_1$	X=0	$F_1(x) = 0$
F2	$F_2(x) = x^T x$	X=0	$F_2(x) = 0$
F3	$F_3(x) = -(15\ 27\ 36\ 18\ 12)x$ $+ x^T \begin{pmatrix} 35 & -20 & -10 & 32 & -10 \\ -20 & 40 & -6 & -31 & 32 \\ -10 & -6 & 11 & -6 & -10 \\ 32 & -31 & -6 & 38 & -20 \\ -10 & 32 & -10 & -20 & 31 \end{pmatrix} x$	$x = (0,11,22,16,6)^T$ $x = (0,12,23,17,6)^T$	$F_3(x) = -737$
F4	$F_4(x) = (x_1^2 + x_2 - 11)^2$ $+ (x_1 + x_2^2 - 7)^2$	$x = (3,2)^T$	$F_4(x) = 0$
F5	$F_5(x) = (9x_1^2 + 2x_2^2 - 11)^2$ $+ (3x_1 + 4x_2^2 - 7`)^2$	$x = (1,1)^T$	$F_5(x) = 0$
F6	$F_6(x) = 100(x_2 - x_1^2)^2$ $+ (1 - x_1)^2$	$x = (1,1)^T$	$F_6(x) = 0$
F7	$F_7(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2$ $+ (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$	$x = (0,0,0,0)^T$	$F_7(x) = 0$

Table 2. Dimension,swarm size and maximum numberof iterations for Test Functions F1-F7

Function	Dim	Swarm Size	Max Iteration
F1	5	20	1000
	10	50	1000
	15	100	1000
	20	200	1000
	25	250	1500
	30	300	2000
F2	5	20	1000
F3	5	150	1000
F4	2	20	1000
F5	2	20	1000
F6	2	50	1000
F7	4	40	1000

problems shown in Table 1[6]. The solution of the equation F(x)=0 except the function F3.. In Table 2 exhibit the swarm's size, the maximum number of iterations as well as dimension for all test functions. For all experiments the initial swarm was taken uniformly distributed inside $[-100,100]^D$,where D is the dimension of the corresponding problem.

In QPSO algorithm, the only parameter setting is Creativity Coefficient β [5],which was gradually decreased for each of the intervals $[1.2, 0.4]$, $[1.0, 0.4]$, $[0.8, 0.4]$ with the number of iterations. And in PSO algorithm, the parameters used for all experiments were $\varphi_1 = \varphi_2 = 2$ and ω was gradually decreased for each of the intervals $[1.2, 0.4]$, $[1.0, 0.4]$, $[0.8, 0.4]$ during the maximum allowed number of iterations. Each of the experiments was repeated 50 runs and the success rate to correct solution as well as the mean number of iterations for each test were recorded.

3.2 Results

The results of PSO and QPSO for the test problems $f_1 - f_7$ are shown in Table 3 and Table 4. Its mean iteration is generated from all tests included incorrect experiments. From the point view of success rate, as shown in Table3, to PSO algorithm, PSO-w is the best choice when ω is gradually from 1.0 to 0.4. Also, to QPSO, β from 1.2 to 0.4 is better than the other two internals as shown in Table 4.

But from the point view of mean iterations, based on the 100 percent success rate, QPSO mostly can reach the correct solution faster than PSO as shown in Table 5. Especially, to test function f1, when dimension is high, the results show that PSO is a better choice, which is because Quantum-behaved PSO algorithm is much fit for global search, especially for higher dimension, and more particles [5].

The convergence graphs for selected test problems are shown in Figure 1, which plot test function value with the number of iteration. As we can see the convergence speed in QPSO is much faster than PSO algorithm.

Table 3. Dimension, Success Rate, Mean Iterations for PSO-w for test F1-F7

F	D	PSO-w					
		w: [1.2,0.4]		w: [1.0,0.4]		w:[0.8,0.4]	
		Succ Rate	Mean Iter	Succ Rate	Mean Iter	Succ Rate	Mean Iter
F1	5	100%	422.27	**100%**	72.8	100%	20.27
	10	100%	434.53	**100%**	90.26	100%	24.77
	15	100%	439.1	**100%**	94.36	100%	25.8
	20	100%	441.67	**100%**	96.3	100%	28.97
	25	100%	653.33	**100%**	99.44	100%	31.23
	30	100%	863.93	**100%**	103.14	100%	34.2
F2	5	100%	423.5	**100%**	77.82	100%	20.22
F3	5	43.3%	718.63	**100%**	125.03	78%	246.56
F4	2	88%	459.44	**100%**	81.24	83.3%	193.36
F5	2	100%	209.64	**100%**	32.9	100%	9.07
F6	2	80%	520.33	**100%**	41.4	56.7%	442.77
F7	4	100%	444.3	**100%**	79.6	100%	34.8

Table 4. Dimension, Success Rate, Mean Iterations for QPSO- β for test F1-F7

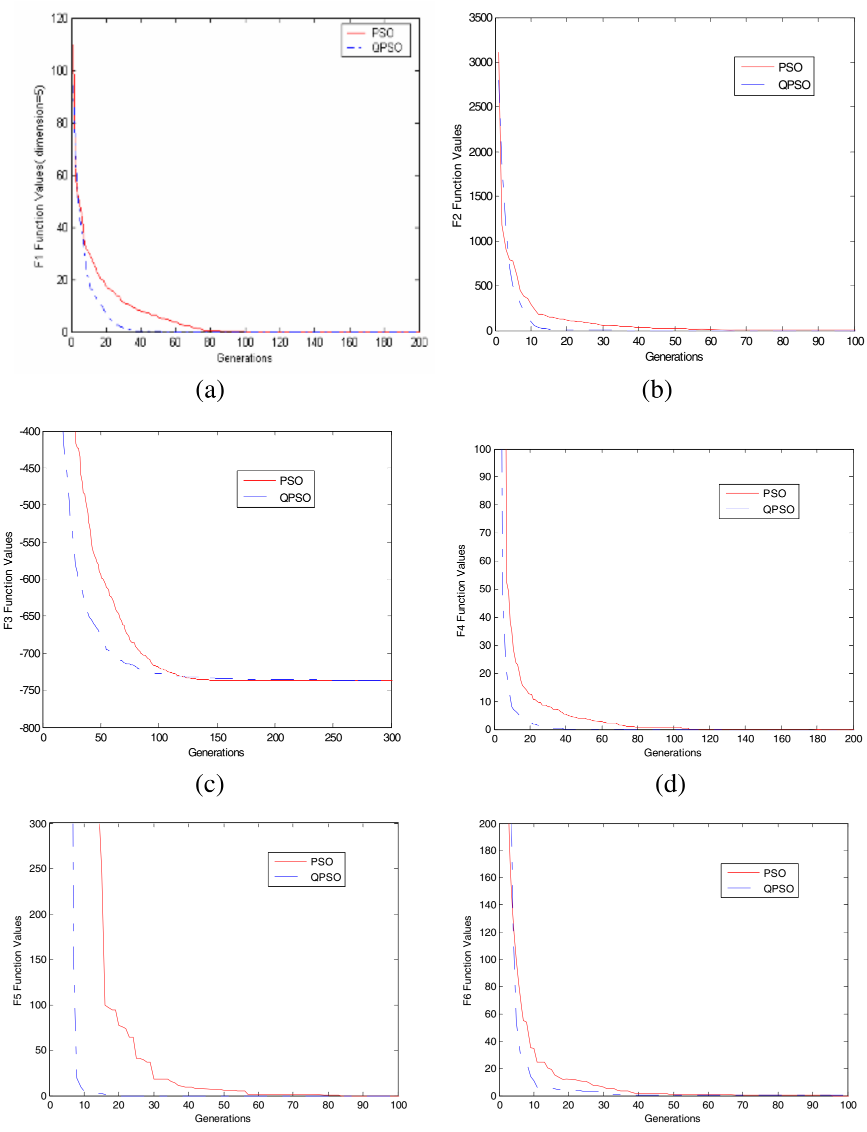
F	D	QPSO-β					
		β: [1.2,0.4]		β: [1.0,0.4]		β: [0.8,0.4]	
		Succ Rate	Mean Iter	Succ Rate	Mean Iter	Succ Rate	Mean Iter
F1	5	**100%**	27.2	100%	21.62	100%	15.14
	10	**100%**	48.26	100%	50.52	100%	27.44
	15	**100%**	64.46	100%	82.26	100%	38.08
	20	**100%**	72.24	100%	120.02	100%	47.74
	25	**100%**	83.86	100%	161.56	100%	58.58
	30	**100%**	92.56	100%	202.44	100%	67.86
F2	5	**100%**	21.2	100%	21.56	100%	15.9
F3	5	**100%**	166.9	84%	284.64	48%	543.9
F4	2	**100%**	19.35	100%	25.8	100%	16.82
F5	2	**100%**	8.95	98%	28.66	98%	28
F6	2	**100%**	14.9	94%	88.14	90%	120.8
F7	4	**100%**	65.7	100%	43.9	100%	33.6

Table 5. Success Rate, Mean Iteration for PSO and QPSO

F	D	PSO-w		QPSO-β	
		w: [1.0,0.4]		β:[1.2,0.4]	
		Succ Rate	Mean Iter	Succ Rate	Mean Iter
F1	5	100%	72.8	100%	27.2
	10	100%	90.26	100%	48.26
	15	100%	94.36	100%	64.46
	20	100%	96.3	100%	72.24
	25	100%	99.44	100%	83.86
	30	100%	103.14	100%	92.56

Table 6. (*Continued*)

F2	5	100%	77.82	100%	21.2
F3	5	100%	125.03	100%	166.9
F4	2	100%	81.24	100%	19.35
F5	2	100%	32.9	100%	8.95
F6	2	100%	41.4	100%	14.9
F7	4	100%	79.6	100%	65.7



(a)

(b)

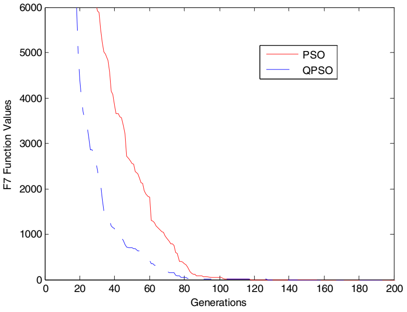
(c)

(d)

(e)

(f)

Fig. 1. Test function value with generations. (a) F1 (b) F2 (c) F3 (d) F4 (e) F5 f) F6 (g) F7.



(g)

Fig. 1. (*continued*)

4 Conclusions

In this paper, we have applied QPSO to integer programming problem. The experiment results on benchmark functions show that QPSO with proper intervals of parameter β can search out the global optima more frequently than PSO, for QPSO can be guaranteed to converge global optima with probability 1 when iteration number $t \to \infty$. Not only QPSO is superior to PSO in this type of problems, but in other optimization problem such as constrained nonlinear program also [11].

Integer programming (IP) is a very important discrete optimization problem. Many of combinatory optimization (CO) can be reduce to IP. Therefore, an efficient technique to solving IP problem can be employed to many CO problems. Based on the work in this paper, which is our first attempt to use QPSO to solve discrete optimization problem, the future work will focus on practicability of QPSO on some NP-complete combinatory problems.

References

1. J. Kennedy, R.C. Eberhart.: Particle Swarm Optimization. Proc of the IEEE InternationalConference on Neural Networks, Piscataway, NJ, USA (1995), 1942-1948
2. Y Shi, R.C.Eberhart: A Modified Particle Swarm Optimizer. Proc of the IEEE Conference on Evolutionary Computation, AK, Anchorage (1998), 69-73
3. M.Clerc, J.Kennedy.: The Particle Swarm: Explosion, Stability and Convergence in a Multi-dimensional Complex Space. IEEE Transactions on Evolutionary Computation (2002), Vol. 6 No.1, 58-73
4. J Sun, B Feng, Wb Xu.: Particle Swarm Optimization with Particles Having Quantum Behavior. IEEE Proc of Congress on Evolutionary Computation (2004), 325-331
5. J Sun, W Xu.: A Global Search Strategy of Quantum-Behaved Particle Swarm Optimization.IEEE conf. On Cybernetics and Intelligent Systems (2004), 111-116
6. K.E.Parsopoulos, M.N.Vrahatis.: Recent Approaches to Global Optimization Problems through Particle Swarm Optimization. Natural Computing, Kluwer Academic Publishers (2002), 235-306

7. D.B.Fogel.: Toward a New Philosophy of Machine Intelligence, IEEE Evolutionary Computation(1995), New York
8. J. Kennedy, R.C. Eberhart.: Swarm Intelligence, Morgan Kaufmann Publishers (2001)
9. D.A.Gall: A Practical Multifactor Optimization Criterion. Recent Advances in Optimization Techniques (1996), 369-386
10. G.Rudolph.: An Evolutionary Algorithm for Integer Programming. Parallel Problem Solving from Nature (1994) 139-148
11. Jing Liu, J Sun, WB Xu.: Solving Constrained Optimization Problems with Quantum Particle Swarm Optimization. Distributed Computing and Algorithms for Business, Engineering, and Sciences (2005) 99-103

Neural Network Training Using Stochastic PSO

Xin Chen and Yangmin Li

Department of Electromechanical Engineering,
Faculty of Science and Technology, University of Macau,
Av. Padre Tomás Pereira S.J., Taipa, Macao S. A. R., P.R. China
ya27407@umac.mo, ymli@umac.mo

Abstract. Particle swarm optimization is widely applied for training neural network. Since in many applications the number of weights of NN is huge, when PSO algorithms are applied for NN training, the dimension of search space is so large that PSOs always converge prematurely. In this paper an improved stochastic PSO (SPSO) is presented, to which a random velocity is added to improve particles' exploration ability. Since SPSO explores much thoroughly to collect information of solution space, it is able to find the global best solution with high opportunity. Hence SPSO is suitable for optimization about high dimension problems, especially for NN training.

1 Introduction

As an attempt to model the processing power of human brain, artificial neural network is viewed as universal approximation for any non-linear function. Up to now many algorithms for training neural network have been developed, especially backpropagation (BP) method. In literatures there are several forms of backpropagation, in which the conventional backpropagation method is the one based on the gradient descent algorithm. Therefore BP is strongly dependent upon the start of learning. That means a bad choice of the starting point may result in stagnation in a local minimum, so that a suboptimum is found instead of the best one. Moreover since many evaluation functions in learning problems are often nondifferentiable or discontinuous in the solution domain, it is difficult to use traditional methods based on derivatives calculations. To overcome these drawbacks, another technology, evolutionary computation, is broadly used in NN training instead of BP.

In evolutionary computation, one well known technology used for NN training is genetic algorithm (GA), which is viewed as a stochastic search procedure based on the mechanics of natural selection, genetics, and evolution. At the same time, since NN training can be viewed as a kind of optimization problem, recently some evolutionary algorithms inspired by social behavior in the nature are also developed to solve NN training, such as particle swarm paradigm, which simulates swarm behavior of ants or birds.

Since particle swarm optimization (PSO) was firstly developed in 1995 [1] [2], it has been an increasingly hot topic involving optimization issues [3,4,5,6].

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1051–1060, 2006.

Roughly speaking, as a recursive algorithm, the PSO algorithm simulates social behavior among individuals (particles) "flying" through a multidimensional search space, where each particle represents a point at the intersection of all search dimensions. The particles evaluate their positions according to certain fitness functions at every iteration, and particles in a local neighborhood share memories of their "best" positions, then use those memories to adjust their own velocities and positions. Due to PSO's advantages in terms of simple structure and easy implementation in practice, there are more and more papers referring training of neural network using PSO. Normally it is accepted that PSO-NN has the following advantages, which are with respect to the drawbacks of BP mentioned above:

- Different from BP, which is normally used in optimization problems with continuous or differentiable evaluation function, PSO algorithm can solve optimization problems with noncontinuous solution domain. Moreover there is no constraint for transfer function, so that more transfer functions, even nondifferentiable ones, can be selected to fulfill different requirements.
- Comparing with BP, the exploration ability embedded in PSO enables NN training using PSO be more efficient to escape from local minima.

But as a stochastic method, PSO method suffers from the "curse of dimensionality", which implies that its performance deteriorates as the dimensionality of the search space increases. To overcome this problem, a cooperative approach named cooperative PSO (CPSO) is proposed in which the solution space is split into several subspaces with lower dimension, and several swarms in these subspaces cooperate with each other to find the global best solution[6] [7]. Cooperating with a traditional PSO, CPSO-H$_K$ can improve performance of PSO significantly. But due to multiple subspaces, there are several times of updating at one iteration, so that generally the computation time of CPSO is several times more than traditional PSO. In fact since in NN training, the dimension of solution space is determined by the number of weights, normally the dimension of solution space is large, so that the computation cost using CPSO becomes large.

We think the reason resulting in "curse of dimensionality" is the fast convergence property of PSO. For a large dimension optimization problem, particles converge so fast that the exploration ability brought by cognitive and social components is weaken too fast to explore solution space thoroughly. Hence instead of splitting solution space into several subspaces, improving exploration ability looks as an alternative to overcome the curse. Therefore we propose a stochastic PSO with high exploration ability to accomplish NN training with relative small size but high efficiency.

2 Stochastic PSO with High Exploration Ability

2.1 Restricts of the Conventional PSO

Given a multi-layer neural network, all its weights are combined together to form a vector which is viewed as a solution vector in a solution space of PSO.

Then a swarm is employed whose members (particles) represent such solution candidates. According to certain criterions, such as minimal root of mean square error (RMSE), all particles congregate to a position on which the coordinate represents the best solution they found.

The conventional PSO updating principle with inertia weight is expressed as follows:

$$\begin{aligned}&v_{id}(n+1)=w_i v_{id}(n)+c_1 r_{1id}(n)(P_{id}^d(n)-X_{id}(n))+c_2 r_{2id}(n)(P_{id}^g(n)-X_{id}(n))\\&X_{id}(n+1)=X_{id}(n)+v_{id}(n+1),\end{aligned} \tag{1}$$

where $d = 1, 2, \cdots, D$, D is the dimension of the solution space, v_i represents current velocity of particle i, $X_i = [\, x_{i1} \quad x_{i2} \quad \cdots \quad x_{iD}\,]^T$ represents current position of particle i. The second part on the right side of updating of $v_i(n+1)$ is named the "cognitive" component, which represents the personal thinking of each particle. The third part is named the "social" component, which represents the collaborative behavior of the particles to find the global optimal solution. Obviously the random exploration ability is determined by $P_i^d(n) - X_i(n)$ and $P_i^g(n) - X_i(n)$. This induces a drawback about PSO exploration that the intension of exploration behavior is totally determined by the rate of decreasing of $P_i^d(n) - X_i(n)$ and $P_i^g(n) - X_i(n)$.

Therefore for a high dimensional optimization problem, such as NN training, when PSO converges quickly, exploration behavior is also weaken so quickly that particles may not search sufficient information about solution space, and they may converge to a suboptimal solution. Since such a relatively low exploration ability is induced by constraints of direction and intension of the cognitive and the social components, a method to overcome the constrained exploration behavior is adding a random exploration velocity to updating principle which is independent on positions. Based on explicit representation (ER) of PSO [8], we propose a new stochastic PSO (SPSO) represented by the following definition.

2.2 Definition of Stochastic PSO (SPSO)

A stochastic PSO (SPSO) is described as follows: Given a swarm including M particles, the position of particle i is defined as $X_i = [\, x_{i1} \quad x_{i2} \quad \cdots \quad x_{iD}\,]^T$, where D represents the dimension of swarm space. The updating principle for individual particle is defined as

$$\begin{aligned}v_{id}(n+1) &= \varepsilon(n)\big[v_{id}(n)+c_1 r_{1id}(n)(P_{id}^d(n)-X_{id}(n))\\&\qquad +c_2 r_{2id}(n)(P_{id}^g(n)-X_{id}(n))+\xi_{id}(n)]\\X_{id}(n+1) &= \alpha X_{id}(n)+v_{id}(n+1)+\tfrac{1-\alpha}{\phi_{id}(n)}(c_1 r_{1id}(n)P_{id}^d(n)+c_2 r_{2id}(n)P_{id}^g(n)),\end{aligned} \tag{2}$$

where $d = 1, 2, \cdots, D$, c_1 and c_2 are positive constants; $P_i^d(n)$ represents the best solution found by particle i so far; $P_i^g(n)$ represents the best position found by particle i's neighborhood; $\phi_i(n) = \phi_{1i}(n) + \phi_{2i}(n)$, where $\phi_{1i}(n) = c_1 r_{1i}(n)$, $\phi_2(n) = c_2 r_{2i}(n)$.

If the following assumptions hold,

1. $\xi_i(n)$ is a random velocity with constant expectation,
2. $\varepsilon(n) \to 0$ with n increasing, and $\sum\limits_{n=0}^{\infty} \varepsilon_n = \infty$,
3. $0 < \alpha < 1$,
4. $r_{1id}(n)$ and $r_{2id}(n)$ are independent variables satisfying continuous uniform distribution in $[0, 1]$, whose expectations are 0.5,

then the updating principle must converge with probability one. Let $P^* = \inf_{\lambda \in (\mathbf{R}^D)} F(\lambda)$ represent the unique optimal position in solution space. Then swarm must converge to P^* if $\lim_n P_i^d(n) \to P^*$ and $\lim_n P_i^g(n) \to P^*$.

2.3 Properties of SPSO

Property 1: Inherent Exploration Behavior
There is a threshold denoted by N_k, such that when $n < N_k$, the individual updating principle is nonconvergent, so that particle will move away from the best position recorded by itself and its neighborhood. But during this divergent process, the particle is still recording its individual best solution and exchanging information with its neighborhood. Hence this phenomenon can be viewed as a strong exploration that during a period shortly after the beginning, i.e., $n \leq N_k$, all particles wander in the solution space and record the best solution found so far. And when $n > N_k$, the swarm starts to aggregate by interaction among particles.

Property 2: Controllable Exploration and Convergence
$\xi(n)$ in SPSO is a stochastic component which can be designed freely. Obviously without the additional stochastic behavior, or $\xi(n) = 0$, the SPSO behaves much like the conventional PSO with relatively fast convergence rate, so that intension of exploration behavior is weaken quickly. To maintain exploration ability, a nonzero $\xi(n)$ is very useful, which makes particles be more efficient to escape from local minima. Moreover in applications $\xi(n)$ with zero expectation is more preferable than nonzero one, because $\xi(n)$ with zero expectation makes particles have similar exploration behavior in all directions.

In the description of SPSO the only requirement of $\xi(n)$ is that its expectation is constant. But there is no restriction about its bound! That implies a very useful improvement that the bound of $\xi(n)$ can be time-varying. If the bound of $\xi(n)$ is constant, as n increases, $\varepsilon(n)\xi(n)$ may be kept relatively strong enough to overwhelm convergence behavior brought by cognitive and social components, so that the convergence of SPSO would be delayed significantly. To overcome this drawback, a time-varying bounded $\xi(n)$ is proposed instead of the constant one, which is expressed as $\xi(n) = w(n)\bar{\xi}(n)$, where $\bar{\xi}(n)$ represents a stochastic velocity with zero expectant and constant value range, $w(n)$ represents a time-varying positive coefficient, whose dynamic strategy can be designed freely. For example the following strategy of $w(n)$ looks very reasonable to balance exploration and convergence behaviors of SPSO.

$$w(n) = \begin{cases} 1, & n < \frac{3}{4}N_b; \\ \eta w(n-1), & n \geq \frac{3}{4}N_b, \end{cases} \tag{3}$$

where N_b represents the maximal number of iterations, and η is a positive constant less than 1. Hence when $n < \frac{3}{4}N_b$, a relatively strong velocity is applied to the particles to increase their exploration ability. And in the last quarter of iterations, the range of $\xi(n)$ deceases iteration by iteration, so that the stochastic behavior brought by $\xi(n)$ will become trivial finally. In a sense during the last part of iterations, such a weakened $\xi(n)$ benefits particles to explore the vicinity around the best solution carefully.

Since SPSO has strong exploration ability than the conventional PSO, it implies that using SPSO, we can accomplish training of NN with relatively fewer particles to reduce computational cost.

2.4 Algorithm of SPSO

Comparing with the conventional PSO, the key improvement of SPSO results from the random exploration velocity $\xi(n)$, which is added to the velocity updating directly. Hence the algorithm of SPSO is very similar to the conventional PSO, which is expressed as the following pseudocode.

```
Encode all weights into a particle coordinate, $X_i = [\, x_{i1} \quad x_{i2} \quad \cdots \quad x_{id} \,]$
Initialize S-PSO
n=1
do
  for $i = 1$ to the swarm size
    Calculate the RMSE, $F(X_i(n))$
    if $F(X_i(n)) < F(P_i^d)$
      $P_i^d = F(X_i(n))$
    end if
  end for
  for $i = 1$ to the swarm size
    $P_i^g = \arg_{j \in \Omega_i} (P_j^d)$
    Determine $w_i(n)$ using (3)
    $\xi_i(n) = w_i(n)\bar{\xi}_i(n)$
    Update $v_i(n+1)$ and $X_i(n+1)$ using (2)
  end for
  $n = n + 1$
while $n <$ the maximal iteration
```

3 Experiment Setup

In order to compare SPSO training with other NN-training algorithms, in this section we propose two tests with respect to two kinds of neural networks. The one is feed-forward version, which is used for a classification problem, the other is recurrent version designed for temporal sequence generation. Four training algorithms, including SPSO, GA, CPSO-H_K, and BP are employed to train NN weights. Both tests are repeated 25 runs, and the average results from 25 runs are accepted as the performance of all algorithms. In Test 1, each algorithm is processed for 1000 iterations (generations), and 2000 RMSE evaluations will be executed in Test 2. The details about the setup of these two tests are presented below.

3.1 Test Setup

Test 1: Feed-Forward NN

A wildly used problem in NN community named as Iris plants problem is employed for the test. Since iris plants include three species, Setosa, Versicolour, and Virginica, species of iris can be classified based on plant measurement, including sepal length, sepal width, petal length, and petal width. A feed-forward neural network with a hidden layer can be employed to accomplish this classification problem, whose architecture is chosen as $4-3-3$ full connected feed-forward neural network (FCFNN), just like Fig. 1 shows. Consequently if we take account of the bias acting on NN nodes, there are 27 weights needing optimization. A differentiable sigmoid function is chosen as transfer function in hidden layer and output layer. A sample set including 150 samples is used as training set. For the test, the root of mean square error (RMSE) is chosen as function evaluation principle, and batch version of training is used to update weights.

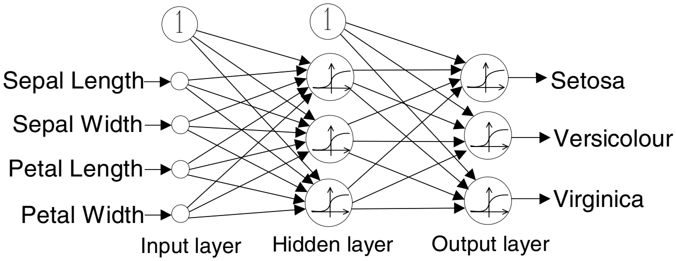


Fig. 1. The feed-forward NN for iris problem

Test 2: Recurrent NN

In this test a full connected recurrent neural network (FCRNN) is trained to generate the following temporal trajectory, whose structure is shown in Fig. 2.

$$y_1^d(t) = 0.35sin(0.5t)sin(1.5t).$$

The discrete-time step is set to $\Delta t = 0.2$, so that if the time range of the trajectories are limited within the interval $(0, 10]$, there are 50 steps within

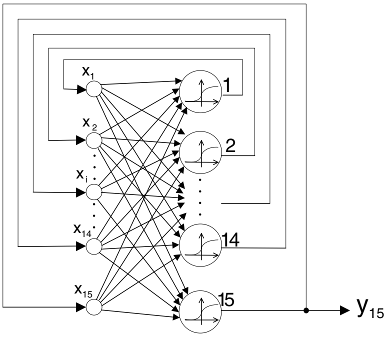


Fig. 2. The recurrent NN for temporal sequence generator

the interval. If there is no external input for FCRNN, the architecture of FCRNN is designed such that 15 nodes in the hidden layer and two output nodes in output layer are employed. Consequently there are 15 × 15 weights needing optimization. The sigmoid function is also chosen as transfer function for all nodes. For the test, the root of mean square error (RMSE) is chosen as function evaluation principle.

3.2 Configurations of All Training Algorithms

In addition to SPSO with nonzero $\xi(n)$, three other algorithms are chosen as comparisons, which are CPSO-H$_K$, GA, and BP, whose configurations are briefly introduced as follows. The swarm size (with respect to SPSO and CPSO) or the number of chromosomes (with respect to GA) for Test 1 is chosen as 25, while that for test 2 is chosen as 100.

- To optimize NN weights using SPSO, all weights are combined to form a potential solution, which is represented by a particle. Then when all particles converge to a position, this position is viewed as the best solution found by SPSO. And the weights picked up from the best solution are the optimized weights for the NN. The parameters used in SPSO are chosen as: $c_1 = c_2 = 3.5$, $\alpha = 0.9$. The form of $\varepsilon(n)$ is of the form $\varepsilon(n) = \frac{5}{(1+n)^{3.5}}$.
- CPSO-H$_K$: The details of CPSO can be found in [7]. The updating principle with decreasing inertia weight is used, where the inertia weight decreases from 0.9 to 0.4 over the search, $c_1 = c_2 = 1.49$.
- GA: A standard GA algorithm with selection, crossover, and mutation operations is employed to optimize NN weights. The crossover probability is set to 0.5, while the mutation probability is chosen as 0.1.
- BP: There are some differences between BPs used in FCFNN and FCRNN. Although both algorithms are based on gradient decent, the BP for FCFNN employs batch version of training, where the learning rate is set to 0.01, while the BP for FCRNN is the epochwise BP through time [9], in which the learning rate is set to 0.3.

4 Test Results

4.1 Test 1: Classification Test Using Feed-Forward NN

This test is proposed to compare the performance of all algorithms for feed-forward NN training. Table 1 shows the test results, including the average RMSE and the standard deviations calculated from 25 runs. RMSE evaluations of all algorithms over iterations (generations) are displayed in Fig. 3.

Table 1. Comparison results of Test 1

Iris Plants Classification		
Algorithm	Average	Standard Deviation
S-PSO ($\xi \neq 0$)	0.2534	0.04483
CPSO-H_5	0.8317	0.4652×10^{-2}
GA	0.5875	0.04869
BP	0.3027	0.08052

From Table 1, we observe that the performance of SPSO has two characters:

- After 1000 iterations, the optimal solution found by SPSO has the least RMSE than other algorithms. Hence SPSO is more efficient to training FCFNN weights.
- Relative to CPSO-H_5, which employs five cooperative swarms, SPSO converges much slowly. This phenomenon implies that the additional random velocity $\xi(n)$ partly counteracts the convergence behavior brought by the cognitive and social components. At the same time the exploration ability of particles in SPSO is enhanced, so that during the optimization, particles have more opportunities to collect information about the space out of convergence trajectories.

A strategy of dynamic $w(n)$ in terms of (3) is applied to $\xi(n)$ to make $\xi(n)$ decrease quickly in the last quarter of iterations. Consequently in the last part of iterations, SPSO converges much quickly, just as Fig. 3 shows, where the trajectory of SPSO decreases quickly in the last quarter of iterations.

4.2 Test 2: Temporal Sequence Generation Using Recurrent NN

Relative to Iris plants classification, temporal sequence generator has a more complex structure about NN, because there are 225 weights to be optimized, while there are only 27 weights in Test 1. Therefore the dimension of solution

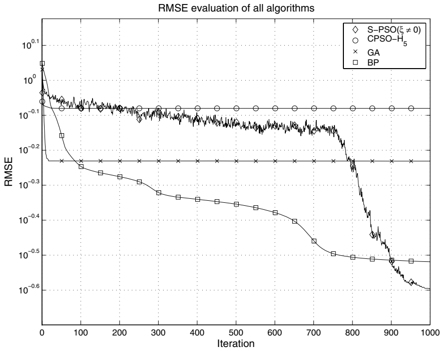
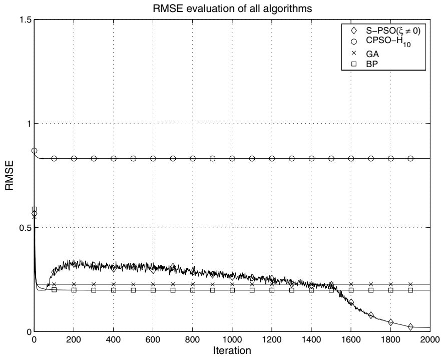


Fig. 3. RMSE evaluation over time in Test 1

Table 2. Comparison results of Test 2

Temporal Sequence Generation		
Algorithm	Average	Standard Deviation
S-PSO ($\xi \neq 0$)	0.01854	0.1911×10^{-2}
CPSO-H_{10}	0.4861	0.6960×10^{-2}
GA	0.2272	0.2708×10^{-2}
BP	0.1987	0.5696×10^{-10}



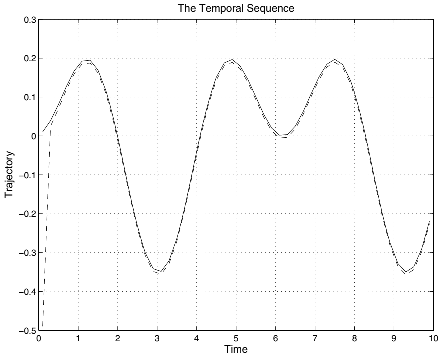


Fig. 4. RMSE evaluation over time in Test 2.

Fig. 5. The temporal sequence generated using SPSO ($\xi \neq 0$)

space for PSOs and GA algorithms is 225, while the number of particles (with respect to SPSO and CPSO) or chromosomes (with respect to GA) is predetermined as 100, which is much lower than the solution dimension. To enhance exploration ability of CPSO, we let 10 cooperative swarms work together.

The results about Test 2 are shown in Table 2, while the RMSE evolution processes of all algorithms are shown in Fig. 4. Obviously the strong exploration ability brought by $\xi(n)$ makes SPSO perform better than other algorithms. CPSO-H_{10} converges too quickly, so that it is premature before particles approach the global best solution. Finally as an example, the trajectory generated by the temporal sequence generator is shown in Fig. 5, which is optimized by SPSO.

5 Conclusion

The paper presents an improved PSO algorithm named SPSO to accomplish NN training. The main improvement about SPSO is that a stochastic exploration velocity denoted by $\xi(n)$ is added to updating principle, so that particles in SPSO have more powerful ability to explore within solution space. Two tests involving FCFNN and FCRNN are proposed to compare SPSO with other algorithms, from which it is observed that although SPSO converges slower than other algorithms, SPSO-NN training performs better than other algorithms.

References

1. Eberhart, R. C., Kennedy, J.: A New Optimizer Using Particle Swarm Theory. Proceedings of the 6th International Symposium on Micro Machine and Human Science. Nagoya, Japan (1995) 39-43
2. Kennedy, J., Eberhart, R. C.: Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Network, Perth, Australia (1995) 1942-1948
3. Juang, C. F.: A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design. IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics, **34**(2) (2004) 997 - 1006
4. Li, Y. and Chen, X.: Mobile Robot Navigation Using Particle Swarm Optimization and Adaptive NN, Advances in Natural Computation, Eds by L. Wang, K. Chen and Y.S. Ong, Springer, LNCS **3612** (2005) 628 - 631
5. Messerschmidt, L. and Engelbrecht, A. P.: Learning to Play Games Using a PSO-Based Competitive Learning Approach. IEEE Transactions on Evolutionary Computation, **8**(3) (2004) 280 - 288
6. Van den Bergh, F. and Engelbrecht A. P.:Training Product Unit Networks Using Co-operative Particle Swarm Optimisers, Proceedings of the IEEE International Joint Conference on Neural Networks, Washington DC, USA (2001) 126 - 131
7. Van den Bergh, F. and Engelbrecht, A. P.: A Cooperative Approach to Particle Swarm Optimization. IEEE Transactions on Evolutionary Computation, **8**(3) (2004) 225 -239
8. Clerc, M. and Kennedy, J.: The Particle Swarm: Explosion, Stability, and Convergence in a Multi-Dimensional Complex Space. IEEE Transactions on Evolutionary Computation, **6**(1) (2002) 58-73
9. Haykin, S.: Neural Networks. Second Edition, Prentice Hall, Upper Saddle River, New Jersey, USA, 1999

Hybrid Training of Feed-Forward Neural Networks with Particle Swarm Optimization

M. Carvalho and T.B. Ludermir

Center of Informatics
Federal University of Pernambuco, P.O. Box 7851
Cidade Universitária, Recife - PE, Brazil, 50732-970
{mrc2, tbl}@cin.ufpe.br

Abstract. Training neural networks is a complex task of great importance in problems of supervised learning. The Particle Swarm Optimization (PSO) consists of a stochastic global search originated from the attempt to graphically simulate the social behavior of a flock of birds looking for resources. In this work we analyze the use of the PSO algorithm and two variants with a local search operator for neural network training and investigate the influence of the GL_5 stop criteria in generalization control for swarm optimizers. For evaluating these algorithms we apply them to benchmark classification problems of the medical field. The results showed that the hybrid GCPSO with local search operator had the best results among the particle swarm optimizers in two of the three tested problems.

1 Introduction

Artificial Neural Networks (ANNs) exhibit remarkable properties, such as: adaptability, capability of learning by examples, and ability to generalize [4]. When applied to pattern classification problems, ANNs through supervised learning techniques are considered a general method for constructing mappings between two data sets: the example vectors and the corresponding classes. As this mapping is constructed the ANN can classify unseen data as one of the classes of the training process.

One of the most used ANN models is the well-known Multi-Layer Perceptron (MLP) [20]. The training process of MLPs for pattern classification problems consists of two tasks, the first one is the selection of an appropriate architecture for the problem, and the second is the adjustment of the connection weights of the network. For the latter is frequently used the Backpropagation (generalized delta rule) algorithm [3], a gradient descent method which originally showed good performance in some non-linearly separable problems, but has a very slow convergence and can get stuck in local minima, such as other gradient-based local methods [12][2]. In this work we focus only on the second task, the optimization of connection weights of MLPs through the use of Hybrid PSO methods.

Global search techniques, with the ability to broaden the search space in the attempt to avoid local minima, has been used for connection weights adjustment or architecture optimization of MLPs, such as evolutionary algorithms (EA) [5], simulated annealing (SA) [21], tabu search (TS) [8], ant colony optimization (ACO) [14] and particle swarm optimization (PSO) [11]. For example: in [4], a genetic algorithm [9] is hybridized with

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1061–1070, 2006.

local search gradient methods for the process of MLP training (weight adjustment); in [1], ant colony optimization is used for the same purpose; in [18], tabu search is used for fixed topology neural networks training; in [19] simulated annealing and genetic algorithms were compared for the training of neural networks with fixed topology, with the GA performing better; in [16] simulated annealing and the backpropagation variant Rprop [15] are combined for MLP training with weight decay; in [22] simulated annealing and tabu search are hybridized to simultaneously optimize the weights and the number of active connections of MLP neural networks aiming classifiers with good classification and generalization performance; in [6], particle swarm optimization and some variants are applied to MLP training without generalization control.

The motivation of this work is to apply the PSO algorithm, its guaranteed convergence variation (GCPSO) and the cooperative PSO (CPSO-Sk) to the process of weight optimization of MLPs. Additionally, we hybridize the first two techniques with the local gradient search algorithm Rprop, combine the cooperative form of the PSO with the guaranteed convergence variation, and employ the GL_5 [13] stop criteria in all the tested algorithms in order to achieve networks with better generalization power. For evaluating all of these algorithms we used benchmark classification problems of the medical field (`cancer`, `diabetes` and `heart`) obtained from the repository Proben1 [13].

The remainder of the article is organized as follows. Section 2 presents the standard PSO and two variations: the Guaranteed Convergence PSO (GCPSO) and the Cooperative PSO algorithms. The experimental setup of this work are described in Section 3. Section 4 presents and analyzes the results obtained from the experiments, and finally, in Section 5 we summarize our conclusions and future works.

2 Particle Swarm Optimization

The PSO optimization technique was introduced by Kennedy and Eberhart in [11] as a stochastic search through an n-dimensional problem space aiming the minimization (or maximization) of the objective function of the problem. The PSO was built through the attempt to graphically simulate the choreography of a flock of birds flying to resources. Later, looking for theoretical foundations, studies were realized concerning the way individuals in groups interact, exchanging information and reviewing personal concepts improving their adaptation to the environment [10].

In PSO, a swarm of solutions (particles) is kept. Let s be the swarm size, n be the dimension of the optimization problem and t the current instant, each particle $1 \leq i \leq s$ has a position $x_i(t) \in \Re^n$ in the solution space and a velocity $v_i(t) \in \Re^n$ which controls the direction and magnitude of its movement. Also, each particle keeps in memory the best individual position $y_i(t) \in \Re^n$ visited until the instant t, and the whole swarm keeps in memory the best position $\hat{y}(t) \in \Re^n$ visited so far by one of its particles.

As the algorithm iterates, the velocity of each particle is determined according to the two main referential points of the search: the <u>individual</u> best position visited so far (cognitive term of the optimization $y_i(t)$) and the <u>global</u> best position visited so far (social term social of the optimization $\hat{y}(t)$). The equations eq. (1) and eq. (2) describe, respectively, how the new velocity and the new position of a particle are determined.

$$v_{ij}(t+1) = w\,v_{ij}(t) + c_1\,r_1(y_{ij}(t) - x_{ij}(t)) + c_2\,r_2(\hat{y}_j(t) - x_{ij}(t)), \quad 1 \leq i \leq s, \quad 1 \leq j \leq n. \tag{1}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad 1 \leq i \leq s, \quad 1 \leq j \leq n. \tag{2}$$

The scalar w is the inertia weight (momentum term) which multiplies the prior velocity of the particle (instant t) and controls the degree of exploration of the search. For a more explorative search it has a value near 1 and for a more exploitative search the value is generally situated near 0.4. The values r_1 and r_2 are uniform random variables taken from $U_{ij1}(0,1)$ and $U_{ij2}(0,1)$, respectively. Both have the role of setting random the influences of the two terms of the equation (cognitive and social). The individual and global acceleration coefficients, $0 < c_1, c_2 \leq 2$, respectively, have fixed and equal values, and are responsible for taking control of how far a particle can move in a single iteration. The best individual position visited so far $y_i(t)$ is updated according to eq. (3), while the best global position visited so far $\hat{y}(t)$ is updated through eq. (4).

$$y_i(t+1) = \begin{cases} y_i(t), & \text{if } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1), & \text{if } f(x_i(t+1)) < f(y_i(t)) \end{cases} \tag{3}$$

$$\hat{y}(t+1) = arg \min_{y_i(t+1)} f(y_i(t+1)), \quad 1 \leq i \leq s. \tag{4}$$

Additionally, the new determined velocity $v_i(t+1)$ is clamped to $[-v_{max}, v_{max}]$, with $v_{max} = x_{max}$ to avoid the "explosion" of the swarm, reducing the likelihood of particles leaving the search space. This does not guarantee that a particle will always be inside the boundaries of the search space, but reduce the distance that a particle will move in one iteration. The standard PSO algorithm is presented in Fig. 1. Rapid convergence in unimodal functions, with good success rate, and premature convergence in multimodal functions are properties frequently attributed to the standard PSO algorithm [6].

```
randomly initialize population of particles
repeat
   for each particle i of the population do
      if $f(x_i(t)) < f(y_i(t))$ then
         $y_i(t) = x_i(t)$
      end if
      if $f(y_i(t)) < f(\hat{y}(t))$ then
         $\hat{y}(t) = y_i(t)$
      end if
   end for
   update velocity and position of each particle according to eqs. (1) and (2)
until stop criteria being satisfied
```

Fig. 1. Standard PSO algorithm

2.1 Guaranteed Convergence PSO

The standard PSO has a property that if $x_i = y_i = \hat{y}$ which means that the particle i is situated on the best point of the search space reached so far, then the velocity update equation (eq. 1) is entirely dependent on the inertia term $w\,v_i(t)$. If the previous velocity of that particle is very close to zero then the particle will stop moving, pushing the particles to that point and causing the premature convergence of the swarm.

A small modification on the standard PSO is made by the guaranteed convergence algorithm (GCPSO) [6] to deal with this problem. The idea is to modify the velocity update equation only for the particles that reached the best global point of the search space to avoid the premature convergence of the swarm and, at the same time, look for better solutions at the vicinity of the current global best position $\hat{y}$. The new equation used is represented by the expression eq. (5) in which i is the index of a particle that reached the current best position of the swarm and $r(t)$ is a random uniform number taken from $U_{ij}(0,1)$. The other particles of the swarm continue to use the standard velocity update equation, i.e. the eq. (1).

$$v_{ij}(t+1) = -x_{ij}(t) + \hat{y}_j(t) + w\,v_{ij}(t) + \rho(t)(1-2r(t)) \tag{5}$$

When this expression is summed to the current position of the particle ($x_{ij}(t)$), we note that the term $-x_{ij}(t) + \hat{y}_j(t)$ resets the current position of the particle to the best global position $\hat{y}(t)$, and the other two terms cause the PSO to perform a random search in the area surrounding the global best position $\hat{y}(t)$. The term $\rho(t)$ of the equation is an adaptive scaling factor that causes this effect on the search performed by the best particle of the swarm. The next $\rho(t)$ value is determined by the expression eq. (6), in which $\#successes$ and $\#failures$ denote the number of consecutive successes and failures of the search in minimizing the objective function, and s_c and f_c are threshold parameters with initial values generally 5. Whenever the $\#success$ counter exceeds the success threshold, this means that the area surrounding the best position may be enlarged leading to the doubling of the $\rho(t)$ value. Similarly, when the $\#failures$ counter exceeds the failure threshold, it means that the area surrounding the global best position is too big and need to be reduced as can be seen in eq. (6).

$$\rho(t+1) = \begin{cases} 2\rho(t) & \text{if } \#successes > s_c \\ 0.5\rho(t) & \text{if } \#failures > f_c \\ \rho(t) & \text{otherwise} \end{cases} \tag{6}$$

Every iteration that the search succeed in minimize the current best position, the $\#successes$ counter is increased and the $\#failures$ counter is reset to zero. In the same way, every iteration that the best global position $\hat{y}(t)$ is not updated, i.e. an unsuccessful iteration, the $\#failures$ counter is increased and the $\#successes$ counter is reset to zero. Every time that the success or failure counters exceed their corresponding thresholds, s_c and f_c, respectively, the exceeded threshold is increased.

2.2 Cooperative PSO

Since the creation of PSO, numerous improvements have been proposed, for example: the use of a constriction factor [6] on the velocity update equation to help ensure convergence of the swarm; the introduction of many topologies of neighborhood other than the standard global one (one best particle for the hole swarm) [10] to prevent the premature convergence of the standard algorithm; the inclusion of the inertia weight (originally not present on the velocity update equation) to control the degree of exploration-exploitation of the search; the binary PSO for binary-domain optimization problems [10]; among others.

Another PSO modification that have been considered with evolutionary algorithms is the use of cooperation between populations. Although competition between individuals usually results in good performance, better improvements can be obtained by the use of cooperation between individuals and, additionally, between isolated populations [6].

One of the first forms of cooperation studied was based on evolution islands [5], where each island corresponds to a geographically isolated subpopulation searching on a separated area of the solution space. After m iterations, the islands send and receive $1 \leq z \leq 5$ individuals promoting an information exchange between the islands. The way islands interact is governed by the topology of the communications. The most used topologies are ring, star, grid, among others. In star topology, for example, every island receive and send individuals from/to the central island.

The use of evolution islands with PSO originated the MultiPSO algorithm which is more appropriated to deal with multimodal problems or unknown-landscape objective functions, due to its capability to keep more diversity than the original PSO. The drawback of the MultiPSO is the additional computational time and memory costs associated with the parallel execution of the isolated PSOs.

Another way of cooperation that have obtained better results in numerical optimization problems is based on the partitioning of the search space, which is applied to problems of high dimensionality. Thus, the original search space of dimension n is divided into $1 \leq k \leq n$ partitions of size d, with $k \times d = n$. In PSO this approach was introduced by the algorithm CPSO-Sk [6,7].

Although the search space of dimension n has been divided into k partitions of dimension d in which a sub-search is executed, the problem remains an n-dimensional one. So, the sub-populations of dimension d need to cooperate offering their best sub-individuals to complete the information necessary to an evaluation of the objective function $f : \Re^n \rightarrow \Re$. Formally, the cooperation between sub-populations is made by the concatenation of the current sub-individual which we want to evaluate and the best sub-individuals obtained so far by the other partitions, at the corresponding positions. This composition is represented by the context vector resultant of the function $b(j, vec)$ expressed on the eq. (7), where vec is the sub-individual of the sub-population j that we want to evaluate and $P_i.\hat{y}$ is the best sub-individual of the sub-population i.

$$b(j, vec) \equiv (P_1.\hat{y}, P_2.\hat{y}, ..., P_{j-1}.\hat{y}, \mathbf{vec}, P_{j+1}.\hat{y}, ..., P_k.\hat{y}) \tag{7}$$

Thus, we can describe the CPSO-Sk algorithm as being realized by the cooperation of PSOs that optimize every one of the k partitions of the search space. The k term is the partitioning factor of the dimension of the problem. For more partitions and more diversity of individuals, we choose a bigger value for the k factor leading to additional computational cost and better results depending on the problem. With $k = 1$, the CPSO-Sk algorithm works exactly as the standard PSO. The CPSO-Sk algorithm generally

Table 1. Parameters of the particle swarm algorithms

Algorithm	Description	Parameter
PSO	swarm size (s)	30
	stop criteria	1000 iterations or GL_5
	quality measure of the MLPs ($f(.)$)	Classif. Error Percentage (CEP)
	search space limit (x_{max})	$[-2.0, 2.0]$
	acceleration factors (c_1 and c_2)	1.4960
	inertia factor (w)	0.7298
GCPSO	initial ρ factor ($\rho(1)$)	1.0
	initial success and failure thresholds (s_c and f_c)	5
CPSO-Sk	partitioning factor (k)	$\lceil 1.3 \times \sqrt{\#weights} \rceil$ [6]

has greater diversity and convergence speed than the standard PSO in a wide variety of problems including the ones with multi-modality.

3 Experimental Setup

The experiments of this work included the standard PSO [11], the guaranteed convergence PSO (GCPSO) [6], the cooperative PSO (CPSO-Sk) [7] and the resilient backpropagation (Rprop) [15] for neural network training. Additionally, we combined the CPSO-Sk algorithm with the guaranteed convergence version of the PSO leading to the CGCPSO-Sk algorithm and hybridized the standard and guaranteed convergence PSOs with the local search operator Rprop leading to the algorithms PSO-Rprop and GCPSO-Rprop. The Rprop local operator was applied with 3 iterations to a particle of the swarm whenever it failed in improving its current individual best position ($y_i(t)$).

In this work also, all of the algorithms tested for weight adjustment used the GL_5 stop criteria [13] for early stopping, in an attempt to increase the generalization power of the networks tested. In the particle swarm algorithms the GL_5 test was evaluated whenever an improvement on the global best position $\hat{y}(t)$ was obtained, and in the Rprop algorithm this test was repeated after m fixed cycles ($m = 5$).

For evaluating all of these algorithms we used three benchmark classification problems of the medical field obtained from the Proben1 repository [13]. The `cancer` data set (9 features and 699 examples) is related to the diagnosis of breast cancer in benign or malignant and is the easiest problem among them. The `diabetes` data set (8 features and 768 examples) is the hardest problem and is related to the detection of diabetes from Pima Indians. Finally, the `heart` data set (35 features) is composed of 920 examples of heart disease prediction. All the three data sets consist of 2-class discrimination problems.

The architecture of the networks was fixed in one hidden layer (number of inputs of the problem - 6 hidden units - 2 output units) as seen on another works on the same data sets [1,4]. The parameters of the PSO algorithms are described on Table 1. It should be noted that: the parameters of the standard PSO are inherited by all the other PSO variants implemented in this work; the GCPSO parameters are inherited by the

Table 2. Mean and standard deviation of the CEP for each algorithm and each of the 3 data sets. The best result for each data set among the Particle Swarm algorithms is indicated in bold.

	Cancer	Diabetes	Heart
PSO	3.6800 (1.5842)	**24.9688** (3.1705)	21.4174 (2.7177)
GCPSO	3.9543 (1.4129)	25.5000 (3.0100)	20.8435 (2.3979)
CPSO-Sk	4.0000 (1.5376)	26.3646 (3.5872)	19.8957 (2.3261)
CGCPSO-Sk	3.7486 (1.4643)	25.8750 (2.7627)	20.3217 (2.3459)
PSO-Rprop	3.8286 (1.5258)	25.3229 (3.2856)	20.0435 (2.2796)
GCPSO-Rprop	**3.6000** (1.2480)	25.9271 (3.1497)	**19.6435** (2.4436)
Rprop	3.4857 (1.4736)	23.5625 (3.1502)	18.9304 (2.4054)

CGCPSO-Sk and GCPSO-Rprop algorithms; and the parameters of the CPSO-Sk are inherited by the CGCPSO-Sk algorithm.

4 Results

In order to compare the Classification Error Percentage (CEP) performance of all the algorithms, each one was executed 50 independent times for each benchmark data set. In every execution, the corresponding data set was randomly partitioned into three subsets: training data set (50%), validation data set (25%) and test data set (25%). The validation data set was directly used by the GL_5 stop criteria in order to avoid the overfitting of the training data by estimating a generalization error for different data.

The results of the experiments are shown in Table 2 in which we report the mean and the standard deviation of the CEP for the 7 tested algorithms with the 3 data sets of the medical field. From the Table 2 it is clear that the `Cancer` and the `Diabetes` are the easiest and the hardest problems, respectively.

For all the tested data sets the local-search specific algorithm Rprop obtained the best results compared to the particle swarm optimizers but with similar performances to the GCPSO-Rprop and PSO algorithms considering the 95% confidence interval of the performed t-tests (Figures 2 and 3). This is an evidence that the GL_5 early

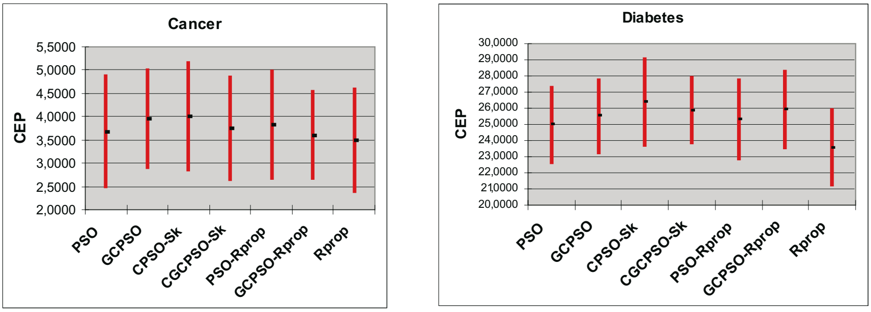


Fig. 2. T-test comparison among the algorithms (CEP) for the data sets `Cancer` and `Diabetes` with 95% confidence interval

stopping heuristic is not appropriated to global search algorithms such as PSO. Also, the hybrid algorithms PSO-Rprop and GCPSO-Rprop obtained similar results than their original counterparts (PSO and GCPSO) as an evidence that the Rprop operator did not considerably improve the PSO and GCPSO generalization performances.

Additionally, from the Table 2, and from the Figures 2 and 3, we note the worse behavior of the cooperative PSOs for NN training in contrast with the very good results obtained for numerical optimization. This is due to the fast convergence caused by the greater diversity of particles created by feature space partitioning which in this work, where the particle swarms tried to minimize only the training error, resulted in some overfitting of the training data. Also, from the t-tests made (Figures 2 and 3), we can statistically state that: all the algorithms had the same performance for the heart problem; all the PSO algorithms had the same performance and the Rprop algorithm was better than the cooperative PSOs for the diabetes problem; and for the heart problem, all the PSO algorithms performed equally and the Rprop algorithm outperformed only the standard PSO.

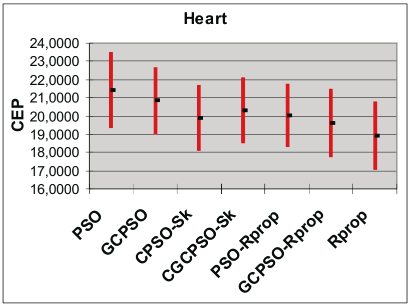


Fig. 3. T-test comparison among the algorithms (CEP) for the Heart data set with 95% confidence interval

5 Conclusions

In this work we have analyzed the feed-forward neural networks training problem with the use of particle swarm optimizers (PSO) and hybrids. Additionally, we have tried to increase the generalization performance of the obtained MLPs by introducing on the PSO algorithm and its hybrids the GL_5 stop criteria [13], guided by the CEP on the validation data set to cease the execution of the algorithms before the overfitting can happen.

The performance of the tested algorithms was evaluated with well known benchmark classification problems of the medical field (Cancer, Diabetes and Heart) obtained from the Proben1 [13] repository of machine learning problems. The results from the performed experiments show that the Rprop specific algorithm had the best performance for the 3 data sets; the GL_5 early stopping criteria, as used here, did not increase the generalization performance of the swarm based algorithms; and that from the particle swarm optimizers, the GCPSO-Rprop obtained the best results for `Cancer` and `Heart` data sets, and the standard PSO for the `Diabetes` data set.

As the results showed in Section 4 the local search operator Rprop used as a kind of mutation did not improve very much the performance of the PSO and GCPSO algorithms. Although, we have so far not tested other local algorithms such as Levenberg-Marquardt and Backpropagation, so we cannot state that the use of local operators in particle swarm optimizers is always of few use.

As future works, we plan to use other heuristics suited for generalization control such as weight decay and pruning. Also, we intend to apply some of the algorithms tested in this work to a more complete neural network optimization methodology based on the simultaneous adjustment of weights and architectures of multi-layer perceptrons (MLP).

Acknowledgments

The authors would like to thank CNPq (Brazilian Research Council) for their financial support.

References

1. C. Blum and K. Socha, "Training feed-forward neural networks with ant colony optimization: An application to pattern classification", *Fifth International Conference on Hybrid Intelligent Systems* (HIS'05), pp. 233-238, 2005.
2. D. Marquardt, An algorithm for least squares estimation of non-linear parameters, J. Soc. Ind. Appl. Math., pp. 431-441, 1963.
3. D. Rumelhart, G.E. Hilton, R.J. Williams, "Learning representations of back-propagation errors", *Nature* (London), vol.323, pp. 523-536.
4. E. Alba and J.F. Chicano, "Training Neural Networks with GA Hybrid Algorithms", K. Deb (ed.), *Proceedings of GECCO'04*, Seattle, Washington, LNCS 3102, pp. 852-863, 2004.
5. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing.* Natural Computing Series. MIT Press. Springer. Berlin. (2003).
6. F. van den Bergh, "An Analysis of Particle Swarm Optimizers", PhD dissertation, Faculty of Natural and Agricultural Sciences, Univ. Pretoria, Pretoria, South Africa, 2002.
7. F. van den Bergh and A.P. Engelbrecht, "A Cooperative Approach to Particle Swarm Optimization", *IEEE Transactions on Evolutionary Computation*, Vol.8, No.3, pp. 225-239, 2004.
8. F. Glover, "Future paths for integer programming and links to artificial intelligence", *Computers and Operation Research*, Vol. 13, pp. 533-549, 1986.
9. J.H. Holland, "Adaptation in natural and artificial systems", University of Michigan Press, Ann Arbor, MI, 1975.
10. J. Kennedy and R. Eberhart, "Swarm Intelligence", Morgan Kaufmann Publishers, Inc, San Francisco, CA, 2001
11. J. Kennedy and R. Eberhart, "Particle Swarm Optimization", in: *Proc. IEEE Intl. Conf. on Neural Networks (Perth, Australia)*, IEEE Service Center,Piscataway, NJ, IV:1942-1948, 1995.
12. K. Levenberg, "A method for the solution of certain problems in least squares", Quart. Appl. Math., Vol. 2, pp. 164-168, 1944.

13. L. Prechelt "Proben1 - A set of neural network benchmark problems and benchmark rules", Technical Report 21/94, Fakultät für Informatik, Universität Karlsruhe, Germany, September, 1994.
14. M. Dorigo, V. Maniezzo and A. Colorni. "Ant System: optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 26, no. 1, pp. 29-41, 1996.
15. M. Riedmiller. "Rprop - description and implementations details", Technical report, University of Karlsruhe, 1994.
16. N.K. Treadgold and T.D. Gedeon, "Simulated annealing and weight decay in adaptive learning: the SARPROP algorithm", *IEEE Transactions on Neural Networks*,9:662-668,1998.
17. R.C. Eberhart and Y. Shi, "Comparison between Genetic Algorithms and Particle Swarm Optimization", Evolutionary Programming VII, Lecture Notes in Computer Science 1447, pp. 611-616, 1998.
18. R.S. Sexton, B. Alidaee, R.E. Dorsey and J.D. Johnson, "Global optimization for artificial neural networks: a tabu search application", *European Journal of Operational Research*(106)2-3,pp.570-584,1998.
19. R.S. Sexton, R.E. Dorsey and J.D. Johnson, "Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing", *European Journal of Operational Research*(114)pp.589-601,1999.
20. S. Haykin, "Neural Networks: A comprehensive Foundation". 2nd Edition, Prentice Hall, 1998.
21. S. Kirkpatrick, C.D. Gellat Jr. and M.P. Vecchi, "Optimization by simulated annealing", *Science*, 220: 671-680, 1983.
22. T.B. Ludermir ; Yamazaki, A. ; Zanchetin, Cleber . "An Optimization Methodology for Neural Network Weights and Architectures" (to be published). *IEEE Transactions on Neural Networks*, v. 17, n. 5, 2006.

Clonal Selection Theory Based Artificial Immune System and Its Application

Hongwei Dai[1], Yu Yang[2], Yanqiu Che[1], and Zheng Tang[1]

[1] Toyama University, Gofuku 3190, Toyama shi, 930-8555 Japan
[2] Tele Electric Supply Service Co. Ltd. Takaoka shi, 939-1119 Japan
dai@hi.iis.toyama-u.ac.jp

Abstract. Clonal selection theory describes selection, proliferation, and mutation process of immune cells during immune response. In this Artificial Immune System (AIS), We select not only the highest affinity antibody, but also other antibodies which have higher affinity than that of current memory cell during affinity mutation process. Simulation results for pattern recognition show that the improved model has stronger noise immunity ability than other models.

1 Introduction

Immune system, one of the most intricate biological systems, is a complex of cells, molecules and organs that has ability to distinguish between self cells and nonself cells[1] [2]. Clonal selection theory [3] describes the basic features of immune response to an antigenic stimulus. It establishes the idea that only those cells that recognize the antigens proliferate. When a B cell recognizes a nonself antigen(Ag) with a certain affinity (the degree of the immune cells recognition with the antigen), it is selected to proliferate and produce antibody(Ab) in high volumes.

Artificial Immune System (AIS), inspired by natural immune system, has been applied for solving complex computation or engineering problems. The authors proposed AIS model to solve multiobjective optimization problems using the clonal selection theory [4] [5]. Other authors [1] [6] described AIS and illustrated the potential of AIS on pattern recognition problem. We also have studied the immune system and proposed different AIS models [7] [8]. In this paper, we built a novel AIS based on the clonal selection theory and simulation for pattern recognition is also performed.

2 Artificial Immune System Model

Based on the *CLONALG* algorithm proposed by Castro L. [9] [10], basic steps of algorithm proposed in this paper can be described as follows:

1. Generate an initial population of antibody randomly, it includes memory population AB_m and reservoir population AB_r.

2. Present an antigen to the system and calculate the affinities between the antigen and all antibodies, based on affinity function.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1071–1078, 2006.

3. Select the highest affinity AB^* and generate a temporary population of clones AB_C.

4. Mutate the clone population. Re-calculate the affinity between mutated clone population and the antigen. Select the antibodies which have higher affinity than current memory cell and regenerate a new antibody as a candidate memory pattern. If its affinity is larger than the current memory pattern, the candidate memory pattern becomes the new memory pattern.

5. Remove those antibodies with low affinities and replace them by new randomly generated members.

6. Repeat step 2-5 until all antigens have been presented.

One generation of the algorithm is complete when all antigens have been presented and all the steps have been carried out for each Ag.

Affinity. Mathematically, either an antibody or an antigen, can be represented by a set of real-valued coordinates $AB = (ab_1, ab_2, ..., ab_M)$ and $AG = (ag_1, ag_2, ..., ag_M)$ respectively. The affinity between antigen and antibody is related to their distance, e.g. the Euclidean distance or the Hamming distance. Equation 1 depicts the Hamming distance D between antibody and antigen.

$$D(j) = H(AG, AB(j)) = \sum_{i=1}^{M} |ag_i - ab_i^j| \quad j = 1, 2, ..., N \tag{1}$$

Obviously, the lower the D, the greater that antibody's affinity with the antigen presented. Hence, we give the affinity $A(j)$ as:

$$A(j) = M - D(j) \quad j = 1, 2, ..., N \tag{2}$$

Selection and Proliferation. According to the affinity, we select the highest affinity AB^* and add it to the clone population.

$$AB^* = argMax\{A(1), A(2), ..., A(N)\} \tag{3}$$

We define CS as the clone population size. Then, the clone population can be indicated as $AB_C = \{AB_C^1, AB_C^2, ..., AB_C^{CS}\}$

Mutation. The mutation of antibody can be implemented in different ways, such as multi-point mutation, substring regeneration and simple substitution [6]. In this paper, the algorithm is implemented by using multi-point mutation. According to the mutation rate *MR*, we randomly select different points and mutate them.

Memory Pattern Update. In the paper [9], the author re-select the highest affinity AB^* for antigen to be a candidate to enter the memory population. If the antigenic affinity of this antibody is larger than the current memory antibody, then mutated AB^* will replace this memory antibody. Refer to Fig. 1, AG is the input antigen. MP is the current memory pattern. Then, we select the highest affinity antibody and generate clones of this antibody and mutate the clone set $(AB_C^1, AB_C^2, ..., AB_C^{CS})$. According to the *CLONALG* rule, the highest affinity antibody AB_C^{CS} is selected as a candidate to enter the memory population. Because the affinity of AB_C^{CS} with antigen is larger than the current memory pattern, the candidate becomes the new memory pattern. The affinity of new memory is 7.

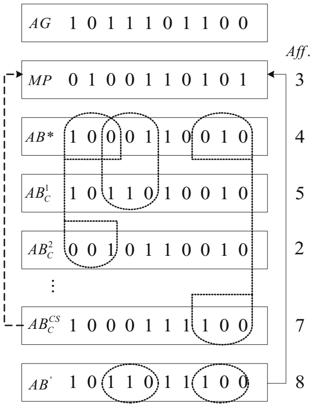


Fig. 1. Antibody mutation and memory pattern update

However, this method perhaps neglects other antibodies which have larger affinity than the current memory antibody, such as AB_C^1. Therefore, we select all these antibodies which have larger affinities than current memory antibody and regenerate a new antibody to enter the memory population. That is, we collect all useful mutation information $(AB_C^1, AB_C^2, ..., AB_C^{CS})$ and regenerate a candidate to enter the memory population. According to the algorithm proposed by us, we can get a new memory pattern AB' which has higher affinity.

3 Simulation

In this section, we evaluate the proposed artificial immune model by being applied to pattern recognition. In order to compare the pattern recognition performance of the proposed model with our early works [7] [8], we select the same data set consists of ten Arabic numberals from website of Carnegie Mellon University [11]. Each pattern is composed of 19*19 pixels.

3.1 Immune Memory

In the following, we will introduce the immune memory process. According to the steps of algorithm proposed in Section 2, we generate random initial antibody population at first. The antibody population includes memory population and reservoir population. We randomly set the elements as 0 or 1.

In order to clearly explain the clonal selection, proliferation, and mutation process, we select the input pattern $'0'$ as an example to illustrate the immune clone processes.

According to Equation (1), (2), we can calculate the affinities of input pattern $'0'$ with all antibodies. Table 1 shows the affinities ($Aff.$) between input pattern and antibodies. We can easily select the highest affinity Antibody AB_M^3 to be proliferated.

Table 1. Affinities of the input pattern $'0'$ with antibodies

	AB_r										AB_m									
Ab	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
$Aff.$	187	183	192	185	187	174	181	176	192	183	174	166	193	178	180	178	190	170	179	167

In clonal selection algorithm, clone population size CS and mutation rate MR are two important parameters. In these papers [9] [12], we find that the authors usually set population size as the value from 50 to 100 and take a very low mutation rate. Here, we set population seize as 100 and the mutation rate is varied between 0.001 and 0.01.

As mentioned in Section 2, the main difference between the proposed algorithm and other models is the update process. *CLONALG* algorithm selects the highest mutated antibody as candidate memory cell. However, in our algorithm, we select all antibodies which have higher affinities than current memory affinity and regenerate a new antibody as candidate memory cell.

According to the mutation rule, we mutate the elements in clone population. Here, a mutation rate of 0.008 is used. The affinity of current memory cell's is 260. The affinities of mutated antibodies are $A(c5) = 261, A(c6) = 261, ..., \mathbf{A(c34) = 263}, ..., A(c93) = 261$.

CLONALG selects the highest affinity antibody as candidate memory cell. If the selected AB's affinity is greater than the current memory cell, then the candidate becomes the new memory cell. Obviously, *CLONALG* selects AB_C^{34} as candidate memory cell and the candidate memory cell becomes new memory cell finally. Fig. 2 (a) shows the new memory cell selected by *CLONALG*. Fig. 2 (b) illustrates the result of the proposed algorithm. This algorithm synthesizes all useful mutations and regenerates a new candidate memory cell. It has a higher affinity than *CLONALG* and becomes new memory cell certainly. Round corners rectangle indicates the mutation area in Fig. 2. It is clearly that the proposed algorithm has higher mutation efficiency than *CLONALG*.

The former example is performed with the mutation rate of 0.008 and population of 100. However, we want to know the exact behaviors of proposed algorithm with different parameters, such as clone population size and mutation rate. We use Hamming distance of all patterns to validate algorithm's performance with different parameters.

Fig. 3 (a) displays the Hamming distance of all patterns, after sufficient mutation, with different population size from 10 to 5000. Here, sufficient mutation means that the system reaches a stable state after certain generations. We find that the system can reach the similar state after sufficient mutation, even if using different clone population size. The lower Hamming distance, the higher

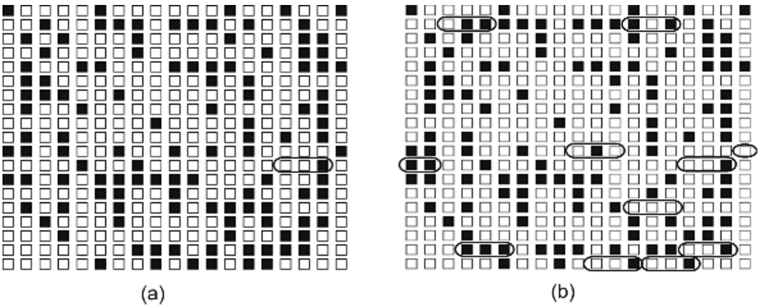


Fig. 2. New memory pattern of *CLONALG* (a) and the proposed algorithm (b)

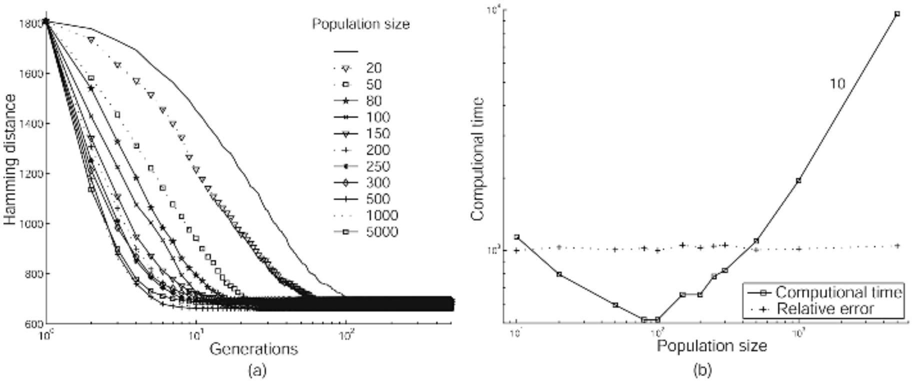


Fig. 3. Hamming distance with different population size

performance the mutated system possesses. Except for the performance of system, the computational cost has to be considered.

Fig. 3 (b) presents the computational time (tick: 1/1000 s) when the system reaching stable state with different population size. The solid line indicates the computational time and the dotted line gives the relative error respectively. It is clearly that the immune system, undergoing different mutation with diverse clone population size, reaches a similar state.

According to the simulation results shown in Fig. 3 (a) and (b), we can draw the conclusion about population size as following: If the clone population size is too low, the system has few possibilities to mutation. On the other hand, if the population size is too large, the computational cost becomes expensive. Based on the simulation results, we consider that the best population size is about 80-100.

Mutation is intended to prevent falling of all solutions in the population into a local optimum of the solved problem. In case of binary encoding we can switch a few randomly chosen bits from 1 to 0 or vice versa.

Fig. 4 presents the simulation results varying process of system with different mutation rate. It is obvious that a higher mutation rate dose not adjust the system to stable state faster than low rate.

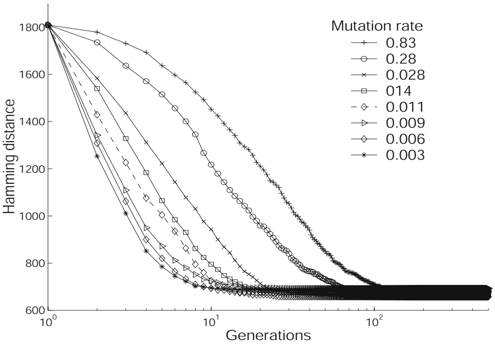


Fig. 4. Hamming distance versus generation with different mutation rate

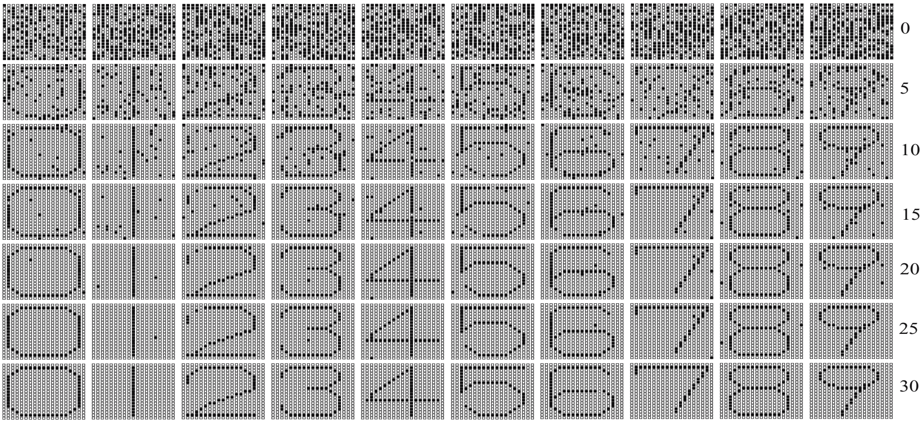


Fig. 5. Mutation process of memory patterns

Based on the former simulation results, we set population size as 100, mutation rate as 0.003 and the mutation process with different generations is shown in Fig. 5.

3.2 Noisy Pattern Recognition

Random noise is added into the input patterns by converting the element's value from 1 to 0 or vice-versa. If the proposed system can map the noisy pattern into the correct category, we think the recognition process is successful. We test different noise number by repeating 1000 times to get a more accurate recognition results. Fig. 6 presents the recognition results with different generations. We find that the lower the clone generation, the lower recognition rate. However, the system seems to reach the stable states when performing about 25 clone generations.

Fig. 7 shows the recognition results of the proposed algorithm with 30 generations. We also present the simulation results of the AIS model [7] [8] proposed

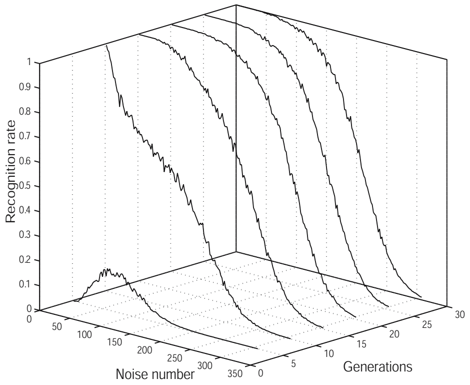


Fig. 6. Recognition rate versus generations

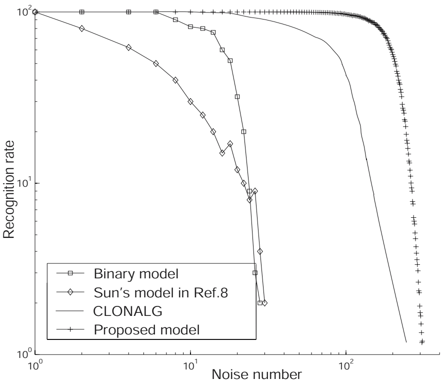


Fig. 7. Recognition results of different AIS models

by us previously. Obviously, the proposed algorithm has a less sensitive to the noise and can recognize the noisy patterns effectively.

4 Conclusions

In this paper, we proposed a clonal selection theory based artificial immune system. In mutation process, we select all mutated antibodies which have higher affinities than current memory cell and regenerate a new candidate memory cell. Simulation results show that the proposed algorithm has an effective mutation performance than *CLONALG*. In order to validate the algorithm, a comparison is performed by applying noisy pattern recognition between the proposed algorithm and other AIS models. Recognition results show that the proposed algorithm has stronger noise immunity and can recognize the unseen noisy patterns more effectively.

References

1. Castro, L., Timmis, J.: Artificial immune system: a novel paradigm to pattern recognition. In: Corchado J, Alonso L, Fyfe C (ed) Proceeding of SOCO, University of Paisley, UK, (2002) 67–84
2. Goldsby, R., Kindt, T., Osborne, B., Kuby, J.: Immunology. W.H. Freeman Co., New York (2003)
3. Burnet, F. The clonal selection theory of acquired immunity. Cambridge press, Cambridge (1959)
4. Wierzchon, S.: Function optimization by the immune metaphor. TASK quarterly. 6(3) (2002) 1–16
5. Coello, C., Carlos, A., Nareli, C.: An approach to solve multiobjective optimization problems based on an artificial immune system. Proceeding of first international conference on artificial immune systems (ICARIS'2002). University of Kent at Canterbury, UK (2002) 212–221

6. Hunt, J., Cooke, D.: Learning using an artificial immune system. Journal of network and computer applications. 19 (1996) 189–212
7. Tang, Z., Hebishima, H., Tashima, K., Ishizuka, O., Tanno, K.: An immune network based on biological immune response network and its immunity. IEICE Trans. Fund. J80A(11) (1997) 1940–1950
8. Sun, W., Tang, Z., Tamura, H., Ishii, M.: An artificial immune system architecture and its applications. IEICE Trans. Fund. E86A(7) (2003)1858–1868
9. Castro, L., Zuben, F.: Learning and optimization using clonal selection principle. IEEE Transactions on evolutionary computation, special issue on artificial immune systems 6(3) (2001) 239–251
10. Castro, L., Zuben, F.: The clonal selection algorithm with engineering applications. Proc. of GECCO'00, workshop on artificial immune systems and their applications, Las Vegas, USA (2000) 36–37
11. URL:http://www-2.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/neural /systems/neocog/
12. Mitchell, M.: An introduction to genetic algorithms. MIT Press (1996)

A Hybrid Algorithm to Infer Genetic Networks

Cheng-Long Chuang[1], Chung-Ming Chen[1], and Grace S. Shieh[2]

[1] Institute of Biomedical Engineering, National Taiwan University,
106, Taipei City, Taiwan
clchuang@ieee.org, chung@ntu.edu.tw
[2] Institute of Statistical Science, Academia Sinica,
115, Taipei City, Taiwan
gshish@stat.sinica.edu.tw

Abstract. A pattern recognition approach, based on shape feature extraction, is proposed to infer genetic networks from time course microarray data. The proposed algorithm learns patterns from known genetic interactions, such as RT-PCR confirmed gene pairs, and tunes the parameters using particle swarm optimization algorithm. This work also incorporates a score function to separate significant predictions from non-significant ones. The prediction accuracy of the proposed method applied to data sets in Spellman *et al.* (1998) is as high as 91%, and true-positive rate and false-negative rate are about 61% and 1%, respectively. Therefore, the proposed algorithm may be useful for inferring genetic interactions.

Keywords: Particle swarm optimization, snake energy model, microarray data, genetic networks.

1 Introduction

The importance of genetic interactions, which often occur among functionally related genes, lies in the fact that they can predict gene functions (Tong *et al.*, 2004). Gaining an understanding of genetic interactions in order to unravel the mechanisms of various biological processes in living cells has been a long-term endeavor. With the emergence of modern biotechnologies, such as advanced microarray technology, inferring genetic interactions among a group of genes has become feasible.

Recently, there have been a few studies on transcriptional compensation (TC) interactions (Lesage *et al.*, 2004; Kafri *et al.*, 2005; Wong and Roth, 2005). Following a gene's loss, its compensatory gene's expression increases, and this phenomenon is known as TC. Reverse transcription (RT)-PCR experiments showed that besides TC, in some cases following a gene's absence, its compensatory gene's expression decreased; we call this phenomenon transcriptional diminishment (TD). Since the mechanism of transcriptional compensation is largely unknown, inferring such interactions is of interest. In particular, interactions among 51 yeast genes which is synthetic sick or lethal (SSL) to SGS1 or RAD27 (Tong *et al.*, 2001; Tong *et al.*, 2004) is of interest. SGS1 (RAD27) has homologues in human cells include the WRN, BLM and RECQ4 (FEN1 and ERCC5) genes. Mutations in these genes lead to

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1079 – 1089, 2006.

cancer-predisposition syndromes, symptoms rSFEMbling premature aging and Cockayne syndrome (Tong *et al.*, 2001 and NCBI database).

With the abundant information produced by microarray technology, various approaches have been proposed to infer genetic networks. Most of them may be classified into three classes: discrete variable models, continuous variable models and graph models. Due to limit of space, our review here is sketchy; for a thorough review, we refer to Shieh *et al.* (2005). Graph models (for example Schäfer and Strimmer, 2005) depict genetic interactions through directed graphs or digraphs instead of characterizing the interactions quantitatively. Some graph models simply reveal structural information, others annotate the directions and signs of the regulations among genes. Owning to the simplicity, graph models usually require much less data than models in the other two categories.

The proposed approach is, in fact, was implemented on indirect interactions among RT-PCR confirmed TC and TD pairs. For ease of description, in this section we henceforth utilize AT and RT, which are direct interactions, to denote TD and TC, respectively. Among RT-PCR confirmed gene pairs, when the time course microarray gene expression (Spellman *et al.*, 1998) of a target gene T is plotted lagged-1 in time behind that of A or R in general, AT gene pairs exhibit similar patterns across time as depicted in the blue zone of Figure 1. On the other hand, RT gene pairs have complementary patterns across time, as illustrated in Figure 2. These motivated us to develop a pattern recognition method that extracts the features of time course microarray data from those confirmed gene pairs, then it can predict similar interactions among genes of interest. Specifically, we generalize the snake energy model (SEM) (Kass *et al.*, 1988) as a shape feature extraction method by incorporating the particle swarm optimization (PSO) algorithm to learn the parameters from data. Moreover, we integrate a simple windowed correlation to the proposed algorithm to improve its discrimination power. We call this method shape feature extraction model (SFEM).

The rest of this paper is organized as follows. Section 2 introduces SFEM and PSO. The proposed SFEM is introduced in Section 3. In Section 4, SFEM is applied to 275 RT-PCR confirmed gene pairs in yeast; the transcriptional compensation interactions are inferred from real microarray data (Spellman *et al.*, 1998). We close with discussion and future directions in Section 5.

2 Model and Methods

2.1 Snake Energy Model

General description. In the field of digital image processing, extracting an object with a designated shape from an image is a major problem needed to be solved. Many image segmentation algorithms have been proposed to accomplish such task. Among them, SEM is a well-known energy-minimizing approach (Kass *et al.*, 1988) used to extract object in an image. In SEM, the model is a manually initialized contour. The idea of SEM is to evolve the contour, subject to constraints from a given gradient

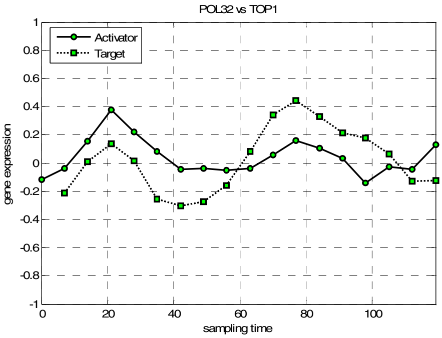


Fig. 1. The gene expression pattern of Activator (POL32) to Target (TOP1) genes across time

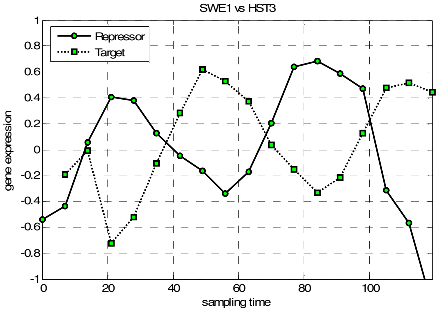


Fig. 2. The gene expression pattern of Repressor (SWE1) to Target (HST3) genes across time

image, and the goodness of the model is decided by a hybridized objective function. For instance, starting with an initial contour around the object to be segmented, the contour is attracted toward its interior normal, and achieves the maximal score once the shape of the contour is well fitted to the boundary of the desired object. Consequently, the contour model dynamically deforms its shape to approximate the contour of the desired object.

The hybridized objective function that drives the shape deformation consists of two forces, which are internal force and external force. The internal force is derived from the shape of the contour model, and the external force is derived from energy distribution of the contour overlap to the image. The function of the internal force is to minimize local curvature of the contour, and the external force is to keep the contour staying on the ridge formed by the gradient image. By applying the hybridized objective function with manual predefined weight factors, SEM can segment to either follow the ridge of the gradient image in a global optimal way, a exceeding precisely way, or any balanced way in between.

The element of the SEM contour is composed of several vertexes. The deformation process of SEM is performed in a number of separately epochs. In each epoch, the goodness regarding to position, velocity, and acceleration of every vertex is evaluated, and the evaluation results in an acceleration of each vertex. The acceleration changes the velocity of the vertex, and the velocity determines the displacement of the vertex in each epoch. After a number of deformation epochs, SEM contour will reach in a converged shape, which means that SEM contour is already achieved an optimal score in the hybridized objective function, and the acceleration and velocity are nearly zero for every vertex in SEM contour. Moreover, the deformation is very robust to local optima problem, because SEM contour deforms segment by segment of the snake. The local-curve fitting technique can capture the finest details of the boundary of interest.

Applying SEM to infer genetic networks: SFEM. In this study, each gene is represented by a node in a graphical model, which is denoted by G_i, where i = 1,

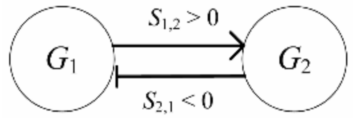


Fig. 3. The graphical model utilized in the proposed method. Genes are represented by nodes, and gene-gene interactions are symbolized by edges between two nodes.

2, …, *N*. The edge $S_{i,j}$ represents the gene-gene interaction between G_i and G_j, where the enhancer gene G_i plays a key role in activating or repressing the target gene G_j. A fundamental graphical model for this work is depicted in Figure 3. Every edge $S_{i,j}$ possesses a value called interaction, which indicates the interaction type and significance level between G_i and G_j. If $S_{i,j}$ is greater than zero, it indicates that the edge is a transcriptional diminishment (TD) interaction, on the other hand, it's a transcriptional compensation (TC) interaction if $S_{i,j}$ is lesser than zero.

In this work, we attempt to discover interaction links $S_{i,j}$ across time by the concept of SEM using gene expression time course data. A new method, called shape feature extraction model (SFEM), is presented to determine the interaction of all possible links between gene pairs. The lagged-1 gene expression of the enhancer gene is treated as SEM contour, and the gene expression of the target gene is considered as the boundary of interest in the SEM technique. We also found that the area surrounded by the lagged-1 expression of the enhancer gene and the expression of the target gene is an essential feature to infer the gene-gene interactions. Therefore, the equations for the SFEM can be formulated as follows:

$$E_{i,j}^{\text{internal}} = \frac{1}{2}\left(\alpha_i \left| \frac{\partial G_i(t')}{\partial t'} \cdot \frac{\partial G_j(t)}{\partial t} \right|^2 + \beta_i \left| \frac{\partial^2 G_i(t')}{\partial t'^2} \cdot \frac{\partial^2 G_j(t)}{\partial t^2} \right|^2 \right) \tag{1}$$

$$E_{i,j}^{\text{external}} = -\sum_{t=t_A-4} \left| \text{corr\_coef}\left(G_i\left(t' : t'+4\right), G_j\left(t : t+4\right)\right)\right|, \tag{2}$$

where $t' = t+1$, $G_i(t')$ is the lagged-1 gene expression level of enhancer gene *i* at time point $t+1$, $G_j(t)$ is the gene expression level of target gene *j* at time point *t*.

In the Equation (1), we use the features of slope and curvature (obtained by the 1st- and 2nd-order partial differential terms) of the gene expression profiles to determine the expression similarity between the lagged-1 expression of the enhancer gene and the expression of the target gene. If these two genes share the same expression pattern across time, the internal force $E_{i,j}^{\text{internal}}$ will result in a positive value because the shapes of two expression profiles are analogous to each other. On the other hand, the internal force $E_{i,j}^{\text{internal}}$ will be in a negative value if two genes are complementary expressed to each other across time.

As to the modified external force $E_{i,j}^{\text{external}}$ in Equation (2), the windowed correlation of expression levels between two genes are summed up into the modified external force $E_{i,j}^{\text{external}}$. By cooperating with windowed correlation, we can simply divide the nonlinear curve into smaller components, and the trend between two genes' expression level can be analyzed using Pearson's correlation. Thus, if the expressions between two genes have the same trend, the external force $E_{i,j}^{\text{external}}$ would appear in positive number; on the other hand, if the expression appears to be compensative, $E_{i,j}^{\text{external}}$ would be in negative number. Therefore, the external force plays a positive role in increasing the true-positive rate of the proposed SFEM.

In this study, because the gene expression profiles are discrete signals, the 1st- and 2nd-order partial differential terms in Equation (3) can be reformulated as follow:

$$\frac{\partial G_i(t)}{\partial t} = \left\| V_{i,t} - V_{i,t-1} \right\| / \Delta t \, . \tag{3}$$

$$\frac{\partial^2 G_i(t)}{\partial t^2} = \left\| V_{i,t-1} - 2V_{i,t} + V_{i,t+1} \right\| / \Delta t \, . \tag{4}$$

where $G_i(t) = V_{i,t} = (t, GE_i(t))$. Therefore, the gene expression profile of gene G_i can be transformed into a 2-D spatial domain (where the horizontal axis represents the time steps, and the vertical axis represents the gene expression levels) to feed to the input of the proposed SFEM algorithm. Hence, the interaction $S_{i,j}$ can be determined as weighted sum of $E_{i,j}^{\text{internal}}$ and $E_{i,j}^{\text{external}}$ as follow:

$$S_{i,j} = \sum_{t=1}^{M} \left[\omega_1 \left(E_{i,j}^{\text{internal}} \right) + \omega_2 \left(E_{i,j}^{\text{external}} \right) \right], \tag{5}$$

2.2 Particle Swarm Optimization

Purpose of utilizing PSO. The main question of most of the existing computational algorithm is: How to determine the parameters of the algorithm? In the classical SEM, the weighting factors ω_1, ω_2, α_i and β_i are still been setup empirically. However, gene expression data obtained from different experiments on different species usually show in different patterns (such as yeast, human, etc). Therefore, to determine a proper set of parameters is needed to yield good prediction results on reconstruction of the genetic regulatory networks.

Variables and methodology. The particle swarm optimization (PSO) algorithm was first introduced by Kennedy and Eberhart (1995). It is a stochastic optimization technique that likely to simulate the behavior of a flock of birds, or the sociological behavior of a group of people. Therefore, PSO is a population based optimization technique.

PSO algorithm contains a population called swarm. All individual particles are distributed in a solution space, and attempt to search for a global optimal solution by sharing the prior experiences obtained at current time. Every time when a particle moves toward to a new position, it would be a seesaw struggle between the optimal solutions obtained by both of the particle itself and the entire population. Generally, all particles will be attracted to the global optimal solution, and in the trending process, other particles will be able to explore new regions, so better solutions can be found by the population.

Let P_i denotes a particle in the swarm population, where $i = 1, 2, \ldots, s$, and s is the size of the total population. The current position, current velocity, individual optimal solution, and global optimal solution are $x_{i,j}$, $v_{i,j}$, $y_{i,j}$ and $\hat{y}_j$, respectively. Therefore, the new velocity of the i-th particle at j-th dimension can be formulated as follow:

$$\begin{aligned} v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_{1,i}(t)\left[y_{i,j}(t) - x_{i,j}(t)\right] \\ + c_2 r_{2,i}(t)\left[\hat{y}_j(t) - x_{i,j}(t)\right] \end{aligned}, \quad (6)$$

where w represents the inertia weight, which typically chosen between 0 to 1 empirically. c_1 and c_2 are weighting factors, which control the dependency of the affecting importance of $y_{i,j}$ and $\hat{y}_j$ in determining the new velocity. $r_{1,i}$ and $r_{2,i}$ are stochastic random factors in the range of (0, 1). The velocities of all particles are usually restricted in the range of ($-v_{\max}$, $v_{\max}$) to prevent those particles from moving too fast, which might miss passing some regions that contain great solutions. The new position of i-th particle is updated by

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1), \quad (7)$$

The individual optimal solution of i-th particle $y_{i,j}$ is renewed using

$$y_{i,j}(t+1) = \begin{cases} y_{i,j}(t) & ,\text{if} \quad f\left(x_{i,j}(t+1)\right) \geq f\left(y_{i,j}(t)\right) \\ x_{i,j}(t+1) & ,\text{if} \quad f\left(x_{i,j}(t+1)\right) < f\left(y_{i,j}(t)\right) \end{cases}, \quad (8)$$

where f denotes the cost function (in this study, f represents the true positive rate using the solution carried by the i-th particle) corresponding to the problem needed to be solved. Finally, the global best solution $\hat{y}_j$ is found by

$$\hat{y}_j(t+1) = \arg\min_{y_{i,j}} f\left(y_{i,j}(t+1)\right),\ 1 \leq i \leq s\ . \quad (9)$$

Thus, in this work, parameters required to be optimized are weighting factors ω_1, ω_2, α_i and β_i in the SFEM algorithm. After several iterations of PSO computation, the global best solutions represent the optimal values of weighting factors can be produced by PSO. Superior in selection of weighting factors can help the proposed SFEM algorithm to learn all essential expression patterns from all dataset of microarray gene expression data using the limited knowledge of RT-PCR confirmed gene-gene interaction pairs.

3 Algorithm

Microarray technology enables scientists to analyze gene-gene interactions on a genomic scale. Large amount of data are provided by a single microarray experiment in the laboratory. However, unfortunately, under different environment settings and diagnosis subject, microarray gene expression data obtained from separately experiments might present in a unique gene expression pattern. Therefore, to recognize the important features in the gene expression pattern becomes an essential task in inferring of genetic networks.

In this study, gene-gene interactions are identified using the interactions produced by SFEM. Let $S_{i,j}$ denotes the interaction between enhancer gene G_i and target gene G_j. The weighting factors ω_1, ω_2, α_i and β_i play essential roles in calculation of $S_{i,j}$, because each of them controls the importance of individual features containing in the expression profile. Therefore, in order to obtain good inferring performance, it is a very important task to select good values to weighting factors ω_1, ω_2, α_i and β_i, and the PSO algorithm was applied to automatic discover a set of suitable parameters to acquire optimal inferring performance subject to a specific microarray gene expression data. A summarized version of the hybrid algorithm of SFEM and PSO is shown as below:

Step 1: initialization A population (swarm) of particles P_i is randomly initialized, where $i = 1, 2, \ldots, s$. Each particle carries a set of solution subject to the weighting factors ω_1, ω_2, α_i and β_i in the SFEM. All properties of the i-th particle, the current position $x_{i,j}$, current velocity $v_{i,j}$, individual optimal solution $y_{i,j}$, and global optimal solution $\hat{y}_j$ are randomly initialized in a limited 4-dimensional solution space.

Step 2: make displacement The new velocity $v_{i,j}$ of each particle at j-th dimension can be calculated, and the new position $x_{i,j}$ can be updated.

Step 3: evaluation Solutions $x_{i,j}$ carried by each particle in the population are applied to the SFEM algorithm for goodness evaluation, and the cost of each particle in the population, represented as $f(x_{i,j})$, is the averaged true-positive rate calculated by leave-one-out cross validation.

Step 4: update individual and global optimal solution Update individual optimal solution $y_{i,j}$ of i-th particle to the current position $x_{i,j}$ if $x_{i,j}$ results in better true-positive rate than $y_{i,j}$. Among all of the individual optimal solution $y_{i,j}$, if any of them has better true-positive rate than global optimal solution $\hat{y}_j$, than update $\hat{y}_j$ by setting $y_{i,j}$ to $\hat{y}_j$.

Step 5: check stop criterion If the max loop hasn't been reached, then return step 2; otherwise, output the optimized solution for all of the weighting factors.

4 Implementation

4.1 Gene Expression Data

Experimental data. Typical indicators of a compensatory relationship genes (paralogues) redundant pathways, and synthetic sick or lethal (SSL) interactions (Wang and Roth, 2005). Out of 51 yeast genes of interest, we focus on 17 genes, which are SSL to *SGS1* or *RAD27* (Tong *et al.*, 2001), whose TC and TD interactions were confirmed by RT-PCR experiments. In this section, SFEM is applied to the cDNA microarray data in Spellman *et al.* (1998) to infer TC and TD interactions. The

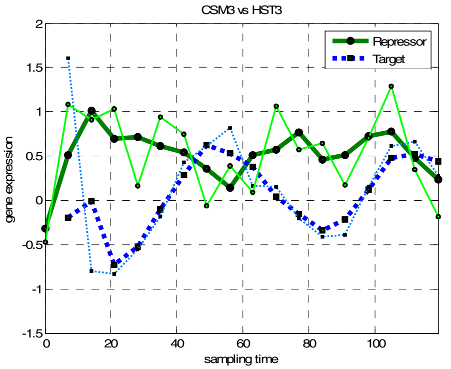


Fig. 4. Effect of a 1 × 3 mean filter applied to the original expression data. Expression levels of repressor gene (CSM3) and target gene (HST3) are depicted in green and blue lines, respectively. Thin lines represent the original expression profile of the repressor and target genes, and the bold ones represent the result after de-noised by the utilized mean filter.

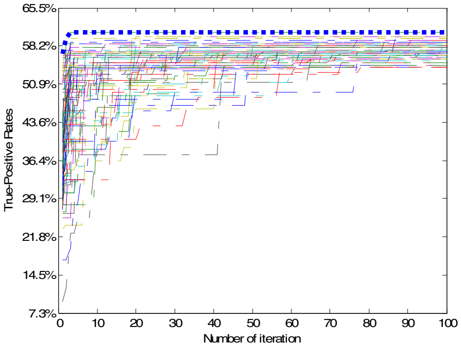


Fig. 5. The PSO evolutionary graph of the cost function (number of predicted true-positive gene pairs) on *Alpha* dataset. Thin dash-dot lines represent evolutionary progress of each particle in the PSO population, and the bold dot line represents best performance obtained at the corresponding iteration. The results are check against 275 gene pairs that are confirmed by RT-PCR experiment.

data used in this paper are results of four experiments. For each experiment, experiment and control group were mRNAs extracted from synchronized by alpha factor, cdc15, cdc28 and elutriation mutants, respectively. There are 18, 24, 17 and 14 sampling time points in each of the experiment with no replicates. The red (R) and green (G) fluorescence intensities were measured from the mRNA abundance in the experiment group and control group, respectively. Log ratios of R to G were used to reconstruct the genetic interactions. A full description of experimental protocol and complete datasets are available at http://cellcycle-www.stanford.edu.

Noises are usually an unavoidable issue in measurement of fluorescence intensities in microarray experiments. At the most of times, noise signals might affect the precision of an analyzing algorithm, and might mislead it to a wrong conclusion on a given dataset. Therefore, in order to suppress noise signals containing in the raw data, a mean filter was applied to the dataset to smooth the variation of the whole gene expression profile. The effect of a 1 × 3 mean filter applied to the original raw data is demonstrated in Figure 4. The original expression profile of repressor gene *CSM3* (thin green line) appears quite noisy. It is very difficult to find any pattern or trend interacting with target gene *HST3* (thin blue line) by eyes observation. Nevertheless, after we applied the mean filter to the expression levels of both genes, both filtered gene expression profiles (bold green and bold blue lines) shows a clearly TC pattern across time. The proposed SFEM infers the genetic networks by analyzing the shape and area formed by each gene pairs, and using a filtered dataset with clearer pattern would benefit to the analysis precision of the proposed algorithm.

4.2 Experimental Results

Results of inferring gene-gene interactions. SFEM was applied to infer the genetic network using *alpha* dataset. The gene-gene influences are represented by the edges interaction links $S_{i,j}$ in the graph model, as shown in Figure 5. By utilizing the PSO algorithm (using PSO parameters w = 0.95, c_1 = 1 and c_2 = 1, which are commonly used empirically), the optimal values for weighting factors of the SFEM can be discovered. The evolutionary graph of the PSO algorithm is plotted in Figure 8, and it shows that the solutions were very quickly been found in no more than 10 iterations using 100 particles in the PSO population, all particles gathered together into 18 local best solution groups. Among all groups, the one that carries the best true-positive rate is the global best solution listed as follows:

$\omega_1 = 1.01$, $\omega_2 = 0.89$, $\alpha_i = 0.24$ and $\beta_i = 2.90$.

Finally, the SFEM was applied to four datasets (*alpha*, *cdc15*, *cdc28* and *elu*) using the weighting factors found by PSO. All of the sampling time points in the datasets are used in the evaluation process without discarding any sampled data (from the 1st to the end time points). And the results yielded by the SFEM using optimized parameters are summarized in the Figure 6, where *TH* is a threshold filter that neutrals the 1st- and 2nd-order partial differential terms in Equation (4) to suppress the level of disruption caused by noise signals. For the *alpha* dataset, the proposed SFEM yielded 63% of true-positive rate with high prediction accuracy (88.4%) when applying an optimized cutoff value. Using *alpha* dataset as training dataset might be the reason of causing good prediction result. However, if we look into the results using the other three datasets, the true-positive rates are still as high as 64% to 66% with prediction accuracies ranging from 72% to 82%.

Results of detecting SSL gene pairs. In this study, the SFEM algorithm was utilized to detect synthetic sick or lethal (SSL) interactions (Wong and Roth, 2005). SSL interactions are important for understanding how an organism tolerates genetic

Dataset	**Leave-one-out cross validation**							
	Cutoff	**Num. of correctly predicted**	**Prediction accuracy**	**Total predicted pairs**	**Num. of correctly detected**	**True-Positive rate**	**Num. of not detected**	**False-Negative rate**
alpha	2.25	43	88.4%	275	172	63%	2	1%
cdc15	1.13	36	66.7%	240	156	65%	0	0%
cdc28	2.94	34	82.4%	240	159	66%	0	0%
elu	1.52	25	72.0%	240	153	64%	4	2%

Fig. 6. The prediction results yielded by the proposed SFEM algorithm. The proposed algorithm was trained by alpha dataset cooperating with PSO algorithm. The trained SFEM was then applied to the other three datasets without further training. The results are check against 275 gene pairs that are confirmed by RT-PCR experiment.

mutations. In present time, the task of identifying SSL interaction in any organism is still far from completion because mapping these networks is highly labor intensive. If the proposed SFEM has the capability of inferring SSL interactions, it could help the biological scientists investigating further SSL interaction with higher efficiency. 13 TC interactions out of 872 SSL interactions were found in Wong and Roth (2005) using microarray data from Hughes *et al.* (2000). The trained SFEM found 177 TC interactions out of 872 SSL interactions using *alpha* dataset. Therefore, we can see that the proposed SFEM algorithm has highly ability of detecting SSL interactions.

5 Discussion

The proposed SFEM learns gene expression patterns from confirmed genetic interactions, confirmed through biological experiments or gathered from databases, then SFEM can predict similar genetic interactions using other not yet seen microarray data. Prediction accuracies of SFEM applied to the *alpha* dataset in Spellman et al. (1998) are as high as 88.4% with true-positive rate 63%. For the *cdc15*, *cdc28* and *elu* datasets, which are excluded from the SFEM algorithm while it was been training. Prediction accuracies yielded by SFEM using *cdc15*, *cdc28* and *elu* datasets are 67%, 82% and 72%, and true-positive rates are 65%, 66% and 64%, which are similar and consistent comparing to the result obtained using *alpha* dataset. Thus, SFEM may be useful for inferring gene-gene interactions using microarray data.

SFEM was also applied to *alpha* dataset again to detect SSL interactions. Among 872 known SSL interactions, a trained SFEM successfully detected 142 SSL interactions. Because mapping SSL networks is an important task to discover how the organisms suffer from gene mutations. Therefore, by applying the proposed SFEM algorithm, larger amount of SSL interactions could be predicted in advance as a guide for scientists to investigate further SSL interactions.

References

1. Chuang, C. L., Chen, C. M. and Shieh, G. S. (2005) A pattern recognition approach to infer genetic networks. *Technical Report C2005-05*, Institute of Statistical Science, Academia Sinica, Taiwan.
2. de la Fuente, A., Bing, N., Hoeschele I. and Mendes P. (2004) Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20:3565-3574.
3. Eberhart, R. C. and Kennedy, J. (1995) A new optimizer using particle swarm theory. *Proc. 6th Int. Symp. Micro Machine and Human Science*, 39-43.
4. Friedman, N., Linial, M., Nachman, I. and Pe'er D. (2000) Using Bayesian network to analyze expression data. *Proc. of the Fourth Annual Conf. on Research in Computational Molecular Biology*, 127-135.
5. Hughes, T. R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton R. et al. (2000) a Functional discovery via a compendium of expression profiles, *Cell*, 102:109-126.
6. Kafri, R. A., Rar-Even and Pilpel, Y., Transcription control reprogramming in genetic backup circuits, *Nat. Genet.*, 37:295-299, 2005.

7. Kass, M., Witkin, A. and Terzopoulos D. (1988) Snake: Snake energy models. *Int. J. Comput. Vision*, 321-331.
8. Kyoda, A., Baba, K., Onami, S. and Kitano, H. (2004) DBRF–MEGN method: an algorithm for deducing minimum equivalent gene networks from large-scale gene expression profiles of gene deletion mutants. *Bioinformatics*, 20:2662-2675.
9. Lesage, G., Sdicu, A. M., Manard, P., Shapiro, J., Hussein, S. *et al.* (2004) Analysis of beta-1,3-glucan assembly in Saccharomyces cerevisiae using a synthetic interaction network and altered sensitivity to caspofungin, *Genetics*, 167:35-49.
10. Schäfer, J. and Strimmer, K. (2005) An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21:754-764.
11. Shieh, G. S., Jiang, Y. C., Hung, Y. C. and Wang, T. F. (2004) A Regression Approach to Reconstruct Gene Networks. *Proc. of 2004 Taipei Symp. on Statistical Genome*, 357-370.
12. Shieh, G. S., Chen, C.M., Yu, C.Y., Huang, J. and Wang, W.F. (2005). A stepwise structural equation modeling algorithm to reconstruct genetic networks. *Technical Report C2005-04,* Institute of Statistical Science, Academia Sinica, Taiwan.
13. Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D.and Futcher, B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Sarcharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9:3273-3297.
14. Tong, A. H. *et al.* (2001) Systematic genetic analysis with ordered arrays of Yeast deletion mutants. *Science*, 294:2364-2366.
15. Tong, A. H. *et al.* (2004) Global mapping of the Yeast genetic interaction network. *Science*, 303:808-813.
16. Wong, S. L. and Roth, F. P. (2005) Transcriptional compensation for gene loss. *Genetics*, published online 5 July, 2005; 10.1534/genetics.105. 046060.

An Intelligent PSO-Based Control Algorithm for Adaptive Compensation Polarization Mode Dispersion in Optical Fiber Communication Systems

Xiaoguang Zhang[1,2], Lixia Xi[1,2], Gaoyan Duan[1,2], Li Yu[1,2], Zhongyuan Yu[1,2],
and Bojun Yang[1,2]

[1] Department of Physics, Beijing University of Posts and Telecommunications,
Beijing 100876, China
{Zhang}zhang.x.g@263.net
[2] Key Laboratory of Optical Communication and Lightwave Technologies,
Ministry of Education, Beijing 100876, China

Abstract. In high bit rate optical fiber communication systems, Polarization mode dispersion (PMD) is one of the main factors to signal distortion and needs to be compensated. Because PMD possesses the time-varying and the statistical properties, to establish an effective control algorithm for adaptive or automatic PMD compensation is a challenging task. Widely used control algorithms are the gradient-based peak search methods, whose main drawbacks are easy being locked into local sub-optima for compensation and no ability to resist noise. In this paper, we introduce particle swarm optimization (PSO), which is an evolutionary approach, into automatic PMD compensation as feedback control algorithm. The experiment results showed that PSO-based control algorithm has unique features of rapid convergence to the global optimum without being trapped in local sub-optima and good robustness to noise in the transmission line that had never been achieved in PMD compensation before.

1 Introduction

In high bit rate optical fiber communication systems, when bit rate is beyond 10Gb/s, polarization mode dispersion (PMD) has become one of the main limiting factors preventing capacity increase, because of PMD induced signal distortion. So PMD compensation has become one of the hot topics in the field of optical communications in recent years [1, 2]. An ordinary feedback type automatic PMD compensator can be divided into three subparts: the PMD monitoring unit, the compensation unit, and the logic control unit. The details of the three subparts will be discussed in Section 2.1. Roughly speaking, the procedure of automatic feedback controlled PMD compensation can be described as: (1) the PMD monitoring unit detects the PMD correlated information as feedback signal. (2) The logic control unit automatically controls the compensation unit by an intelligent and rapid control algorithm through analyzing the feedback signal, and the compensation completes as result. In [1] and [2] the algorithm used as the control part of a PMD compensator employed gradient based peak

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1090 – 1100, 2006.

search methods. However, we found that as the numbers of control parameters increased, the gradient based algorithm often became locked into local sub-optima, rather than the global-optimum. Besides, it would be less effective for a system with a relatively high noise level in the PMD monitor, because the gradient information between neighboring signals would be submerged in noise. We introduced the particle swarm optimization (PSO) into logic control unit for the adaptive PMD compensator for the first time, and realized a series of compensation experiments. With PSO algorithm we give some of the feasible and effective solutions for some critical problems that were headaches in the field of PMD compensation for a long time past.

2 A Brief Introduction to Polarization Mode Dispersion and PMD Compensation

2.1 Polarization Mode Dispersion

Polarization mode dispersion has its origins in optical birefringence [3]. In a single mode fiber, an optical wave traveling in the fiber can be represented as the linear superposition of two orthogonal polarized HE_{11} modes. In an ideal fiber, with a perfect circular symmetric cross-section, the two modes HE_{11}^{x} and HE_{11}^{y} are indistinguishable (degenerate) with the same time group delay. However, real fibers have some amount of asymmetry due to imperfections in manufacturing process or mechanical stress on the fiber after manufacture as shown in Fig. 1. The asymmetry breaks the degeneracy of the HE_{11}^{x} and HE_{11}^{y} modes, resulting in birefringence with a difference in the phase and group velocities of two modes.

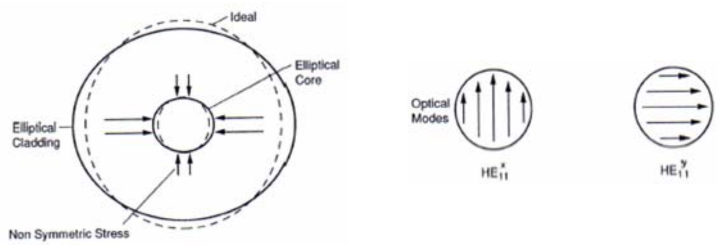


Fig. 1. Asymmetry of a real fiber and degeneracy of two orthogonal HE_{11} modes

If a pulsed optical wave that is linearly polarized at 45° to the birefringence axis is launched into a birefringent fiber, the pulse will be splitted and separated at output end of the fiber due to the different group velocities of two HE_{11} modes, as shown in Fig. 2, resulting in a signal distortion in optical transmission system. The time separation between two modes is defined as differential group delay (DGD) $\Delta\tau$. Roughly speaking, the fast and slow axis is called principal states of polarization. This phenomenon is called polarization mode dispersion.

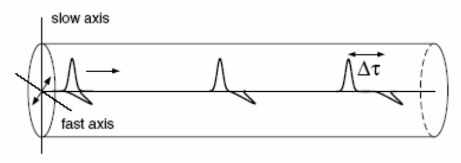
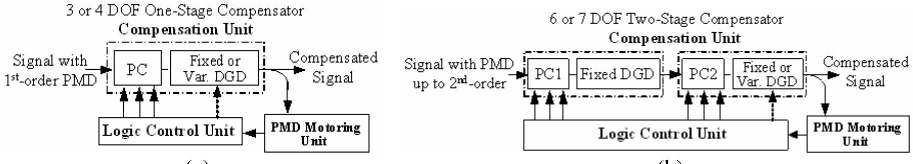


Fig. 2. Pulse splitting due to birefringence

2.2 The Configuration of Automatic PMD Compensator

Polarization mode dispersion can be divided into first-order and high-order PMD according to its Taylor-series expansion with frequency deviation $\Delta\omega$ from the carrier frequency ω_0. The first-order and second-order PMD are the two dominant impairment factors to the optical fiber transmission systems.



(a) (b)

Fig. 3. The configuration of one-stage and two-stage compensators

There are two compensation schemes, pre-compensation and post-compensation. As mentioned in the section of Introduction, for the scheme of optical feedback post-compensation, the compensator has three subparts: the PMD motoring unit, the compensation unit, and the logic control unit. It is widely believed that the one-stage compensators are able to compensate PMD to the first-order. For the one-stage compensator, the compensation unit is composed of a polarization controller (PC) whose function is to transform the state of polarization (SOP) of input optical wave into output state, and a differential group delay (DGD) line with the purpose of eliminating the DGD of the input optical signals (Fig. 3 (a)). One-stage compensator have 3 or 4 control parameters (or degrees of freedom (DOF)), three for PC and one for DGD line, to be controlled depending on whether the DGD line is fixed or varied. The two-stage compensators, composed of two segments of PC+DGD, can compensate the PMD up to the second-order [4]. They have two compensation units and 6 or 7 control parameters (or DOF) to be controlled depending on whether the second delay line is fixed or varied (Fig. 3 (b)).

The automatic PMD compensation is a process for a control algorithm to find an optimal combination of control parameters, in order for the feedback signal (PMD motoring signal) to reach a global optimum, in an intelligent, fast, and reliable manner. In our experiment, the degree of polarization (DOP), obtained by an in-line polarimeter in the PMD monitoring unit, was used as feedback signal. The DOP of light wave is defined as follows using Stokes parameters S_0, S_1, S_2, S_3.

$$\mathrm{DOP} = \frac{\sqrt{S_1^2 + S_2^2 + S_3^2}}{S_0} \tag{1}$$

The DOP of any light wave varies in the range of 0 to 1. DOP takes value of 1 when the light wave is completely polarized, 0 for unplolarized light, takes the value between 0 and 1 when the light wave is partially polarized. PMD would make a complete polarized light signal in fiber to become partially polarized or even unpolarized. The optical pulses at the receiving end have a DOP of 1 when there is no PMD in the fiber link, and the DOP value decreases as PMD increases [2]. The polarization controller used in the compensation unit is the electrically controlled whose three cells were adjusted by controlling voltages in the experiment. In our experiment, fixed delay line was adopted. Therefore the control parameters for the one-stage compensator were 3 voltages (V_1, V_2, V_3) of PC, and the control parameters for the two-stage compensator were 6 voltages (V_1, V_2, V_3, V_4, V_5, V_6) of PC1 and PC2.

The procedure of the PMD compensation is: the control algorithm in logic control unit automatically adjusts 6 voltages (V_1, V_2, V_3, V_4, V_5, V_6) of PC1 and PC2 until the feedback signal DOP reaches its maximum.

3 Automatic PMD Compensation Using PSO Algorithm

3.1 The Theory of PSO-Based Control Algorithm

As mentioned in Section 2.2, the DOP value that is taken as the feedback signal decreases as PMD in the fiber link increases. In the feedback post-compensator, the task of the control algorithm in logic control unit is automatically searching for global maximum DOP through adjusting the multi-voltages of the polarization controllers (PCs) in the compensation unit, in an intelligent, fast, and reliable manner, which can be described mathematically as:

$$\underset{parameters}{\mathrm{MAX}}\,(function) \tag{2}$$

where the *function* in bracket represents the DOP value in the fiber link. The *parameters* here are the voltages for controlling the PCs in the compensation units. There is no simple method to predict function (2) in an automatic compensation system. A good algorithm is, therefore, required to solve problem (2), which is the problem of searching for the global maximum of DOP in a multi-dimensional hyperspace. The number of parameters (or degree of freedom) is the number of dimensions of the hyperspace, and is 3 for our one-stage compensator and 6 for our two-stage compensator.

Generally, more degrees of freedom result in more sub-maxima existing, which will increase the hard task of the searching algorithm. Unfortunately there exist several DOP sub-maxima in the compensation process. Fig. 4 is a typical DOP surface map for our PMD compensation system.

We can see in Fig. 4 that, there are several sub-maxima beside a global maximum in the searching space. We can also find that, the DOP surface is not smooth because of the noise in the fiber link.

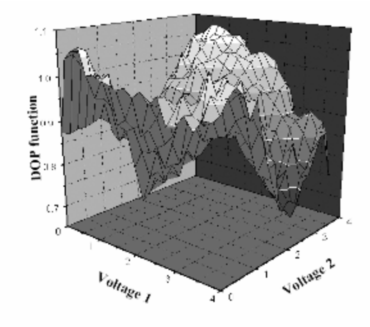


Fig. 4. The DOP surface map in the PMD compensation system

In most of the related literature about PMD compensation, the adopted control algorithms have not been explicitly characterized. In [1] and [2] the algorithm used employed gradient based peak search methods. However, with the numbers of control parameters increasing, the gradient based algorithm often became locked into local sub-maxima, rather than the global-maximum. Besides, it would be less effective for a system with a relatively high noise level as shown in Fig.4, because the gradient information between neighboring signals would be submerged in noise. Therefore finding a practical feedback control algorithm with the desirable features is still a challenging task. A competitive searching algorithm in PMD compensation should at least satisfy following features: (1) rapid convergence to the global optimum rather than being trapped in local sub-optima; (2) good robustness to noise.

The PSO algorithm, proposed by Kennedy and Eberhart, has proved to be very effective in solving global optimization for multi-dimensional problems in static, noisy, and continuously changing environments [5, 6]. We introduced for the first time the PSO technique into automatic PMD compensation in a series of experiments [7].

At the beginning, the PSO algorithm randomly initializes a population (called swarm) of individuals (called particles). Each particle represents a single intersection of multi-dimensional hyperspace. The position of the *i*-th particle is represented by the position vector $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$. In the D-dimensional-DOF PMD compensation scheme depicted in Fig.3, the components of the *i*-th particle are represented by the combination of D voltages (V_1, V_2, …, V_D). The particles evaluate their position relative to a goal at every iteration. In each iteration every particle adjusts its trajectory (by its velocity $V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$) toward its own previous best position, and toward the previous best position attained by any member of its topological neighborhood. If any particle's position is close enough to the goal function, it is considered as having found the global optimum and the recurrence is ended.

Generally, there are two kinds of topological neighborhood structures: global neighborhood structure, corresponding to the global version of PSO (GPSO), and local neighborhood structure, corresponding to the local version of PSO (LPSO). For the global neighborhood structure the whole swarm is considered as the neighborhood (Fig.5 (a)), while for the local neighborhood structure some smaller number of adjacent members in sub-swarm is taken as the neighborhood (Fig.5 (b)) [8]. The detail of process for implementing the global version of PSO can be found in [9]. In the global neighborhood structure, each particle's search is influenced by the best position found by any member of the entire population. In contrast, each particle in the local neighborhood structure is influenced only by parts of the adjacent members. Therefore, the local version of PSO (LPSO) has fewer opportunities to be trapped in sub-optima than the global version of PSO (GPSO).

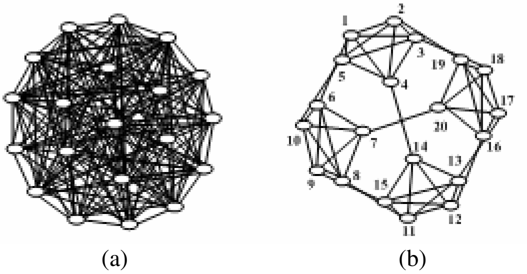


Fig. 5. One of two topologic structures for (a) global neighborhood and (b) local neighborhood

Generally, the larger the number of particles adopted in PSO, the fewer the opportunities to be trapped in sub-optima, but the greater the time spent searching for the global optimum. In our experiment 20 particles are used either in GPSO or LPSO, which is a balance between the accuracy required in searching for the global optimum and time consumed. For LPSO neighborhood, it is found that having 5 neighbors for every particle gives the highest success rate in finding the global optimum (Fig.5 (b)) [8]. The relationship for structure of LPSO we adopted is labeled in Table 1.

Table 1. The neighborhood structure for topologic shown in Fig. 5 (b)

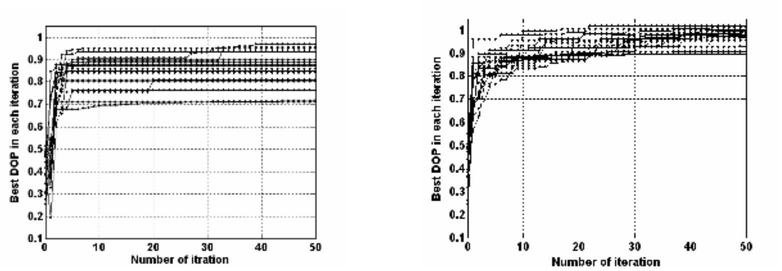
Particle	Neighbors	Particle	Neighbors
1	2,3,4,5,6	11	8,12,13,14,15
2	1,3,4,5,19	12	11,13,14,15,16
3	1,2,4,5,18	13	11,12,14,15,17
4	1,2,3,5,14	14	4,11,12,13,15
5	1,2,3,4,10	15	9,11,12,13,14
6	1,7,8,9,10	16	12,17,18,19,20
7	6,8,9,10,20	17	13,16,18,19,20
8	6,7,9,10,11	18	3,16,17,19,20
9	6,7,8,10,15	19	2,16,17,18,20
10	5,6,7,8,9	20	7,16,17,18,19

3.2 The Results of the Automatic PMD Compensation Using PSO

In the series of our automatic PMD compensation, we will describe here the results of one experiment we have done, the automatic second-order PMD compensation using two-stage compensator in 40Gb/s time-division-multiplexing (OTDM) transmission system, in order to show the effectiveness by using PSO algorithm. We employed both GPSO and LPSO as the control algorithm respectively, in order to make a comparison of effectiveness of them.

We conducted 18 times of compensation experiments, whose setup is depicted in Fig.3 (b), by controlling 6 voltages of PC1 and PC2 through the GPSO and LPSO algorithms, respectively. We randomly selected the 18 different initial PMD states of the PMD emulator (corresponding to 18 different initial DOP values) for 18 different experiments. The function of PMD emulator, which is located in front of the compensator, is to emulate PMD as same as in real fiber. In every process of global DOP maximum searching, we recorded the variation of best DOP values in each iteration and, with the maximum iteration number set to 50, the results are shown in Fig. 6.

Because of more local sub-maxima and relative high level noise in 6-DOF system, for the GPSO case there are some initial PMD states for which DOP only achieves the value of 0.7 (Fig. 6(a)), corresponding to being trapped in local sub-maxima and failure of compensation. In contrast, for the LPSO case all final searched DOP values exceed 0.9, no matter what the initial PMD state is (Fig. 6(b)). Furthermore, if we set DOP value of 0.9 as the criterion which is considered to achieve the compensation, all the DOP values reach that criterion within about 25 iterations, corresponding to compensation time less than 550 ms. We can draw the conclusion that LPSO can better undertake the task of solving multi-dimensional problems, and that it is a better searching algorithm for adaptive PMD compensation up to high-order.



(a)

(b)

Fig. 6. The best DOP vs. iteration recorded in 6-DOF second-order PMD compensation using LPSO (a) and GPSO algorithm (b)

Fig. 7 shows the eye diagrams displayed on the screen of the oscilloscope at receiver end, in the whole procedure of automatic PMD compensation in 40Gb/s OTDM optical transmission system. The eye diagrams in left column of Fig. 7 are the 40Gb/s OTDM signals in situations of back-to-back, before and after PMD compensation. The eye diagrams in right column are the 10Gb/s demultiplexed signals with the same meaning. When we adjusted the PMD emulator with the result that the eyes were closed, implying severe PMD induced signal distortion with DOP = 0.23. After switching on the compensator, the eyes opened and DOP reached close to 1 within about 500 milliseconds through optimum searching by LPSO algorithm.

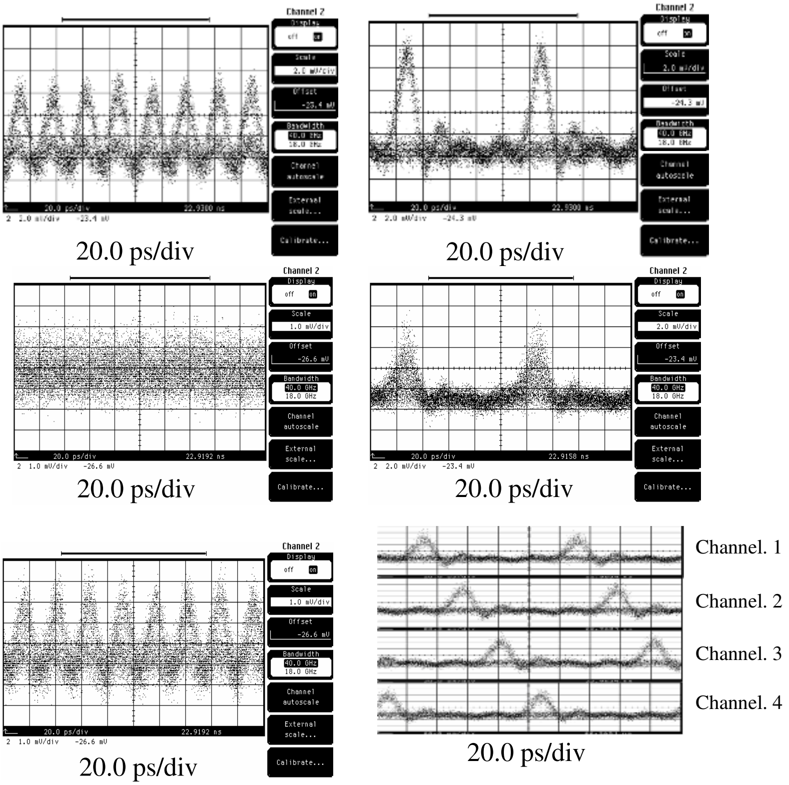


Fig. 7. Eye diagrams to show the procedure of automatic PMD compensation in 40Gb/s OTDM optical transmission system. (a) Back-to-back 40Gb/s OTDM signal. (b) Back-to-back demultiplexed 10Gb/s signal. (c) 40Gb/s signal without PMD compensation. (d) Demultiplexed 10Gb/s signal without PMD compensation. (e) 40Gb/s signal with PMD compensation. (f) Four demultiplexed 10Gb/s signals with PMD compensation.

3.3 The PSO Technique Used in Tracking Process of the Control Algorithm

Because the PMD in the real fiber link always randomly changes due to changes in the environment such as temperature fluctuations etc, the tasks of the PMD compensator are not only to quickly recover the optical fiber communication system from bad state when PMD severely distorts the optical transmission signals, but also to endlessly maintain this recovered state unchanged in a real-time manner. Therefore, the algorithm for real-time adaptive PMD compensation should include two stages. First, the searching algorithm finds the global optimum from any initial PMD condition. Then the tracking algorithm starts to track the changed optimum.

From our experiment, when the PMD in the fiber link changes, the global DOP maximum just drifts away from the previous location as shown in Fig. 8. A natural thought of solution is a tracking method of slight disturbances or dithering around the previous DOP maximum as shown Fig. 9, which was adopted in [2]. This was also gradient-based control algorithm which would not adequate for the systems with a relatively high noise level in the PMD monitoring unit. Furthermore, for a one-DOF control system, there are two directions (positive and negative) for dithering. For a two-DOF system, there will be 8 directions (east, west, south, north, southeast, southwest, northeast, northwest), and for D-DOF, 3^D-1 directions. In conclusion, for multi-DOF systems the amount of calculation will become comparatively large, making it unsuitable for real-time tracking.

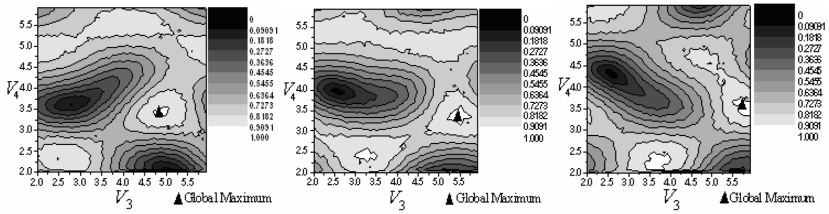


Fig. 8. Location drifting of global DOP maximum from (a) to (c) indicating the PMD changes with time

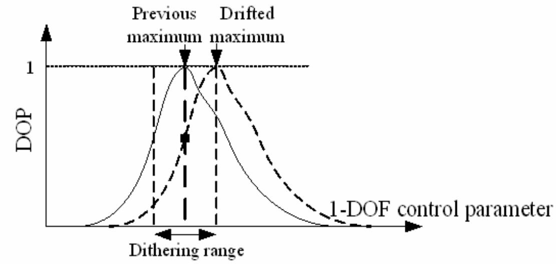


Fig. 9. The dithering solution for tracking the varied DOP maximum

Because of its good performance in the presence of noise, and its multi-dimensional searching capability, we used the PSO searching technique in the smaller 6-dimensional local space around the previous optimum location to achieve the goal of tracking changing optimum. After the global optimum search process is completed, the tracking algorithm starts to work according to the DOP values. When the DOP in the fiber link is higher than 98% of that obtained for the previous optimum, the algorithm does nothing. Otherwise, as long as the DOP is lower than this criterion, local space searching is initiated. The size of the local searching space is adjusted with time according to the deviation from the criterion DOP, which is set to 0.9×98%=0.88 for the experiment. For the tracking algorithm, 5 particles and GPSO were adopted because of the faster speed needed for tracking and the smaller space in which to search. The flow chart of the control program is shown in Fig. 10.

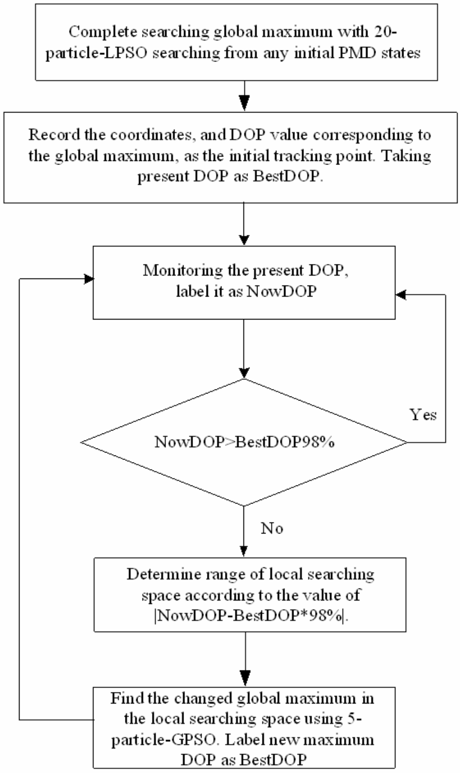

Complete searching global maximum with 20-particle-LPSO searching from any initial PMD states
Record the coordinates, and DOP value corresponding to the global maximum, as the initial tracking point. Taking present DOP as BestDOP.
Monitoring the present DOP, label it as NowDOP
NowDOP>BestDOP98%
Yes
No
Determine range of local searching space according to the value of |NowDOP-BestDOP*98%|.
Find the changed global maximum in the local searching space using 5-particle-GPSO. Label new maximum DOP as BestDOP

Fig. 10. The flow chart of the control program based on PSO

In the experiment, the tracking algorithm worked well when the PMD in the fiber link varied slowly and smoothly with the environment. The eye diagrams are nearly unchanged. Fig.11 shows the tracking results with small vibration of DOP values around the criterion (0.88). But if there is a sharp disturbance in the fiber link, the tracking algorithm will force the system rapidly to recover to the condition beyond criterion.

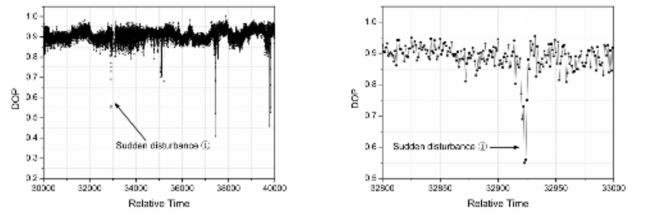


Fig. 11. The performance of the tracking algorithm for tracking the changed optimum DOP. (a)In relative long time, there are some sudden disturbances by sudden rotating the PC of emulator. (b)Details of sudden disturbance ①.

4 Conclusions

For the first time, we have introduced the particle swarm optimization into automatic polarization mode dispersion compensation. The experiment showed that PSO exhibited the desirable features for automatic PMD compensation of rapid convergence to the global compensation optimum searching without being trapped in local sub-optima that corresponded to the failure of compensation, and good robustness to noise in the transmission line. However, all these problems that PSO can solve were headaches in the field of PMD compensation for a long time past. By comparison of global version of PSO (GPSO) and local version of PSO (LPSO), it was shown that LPSO is better solution for automatic PMD compensation.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 60577064, and the National "863" High Technology Project of China under Grant 2001AA122041.

References

1. Noé, R., Sandel, D., Yoshida-Dierolf, M., Hinz, S., Mirvoda, V., Schöpflin, A., Glingener, C., Gottwald, E., Scheerer, C., Fischer, G. Weyrauch, T., Haase, W.: Polarization Mode Dispersion Compensation at 10, 20, and 40Gb/s with Various Optical Equaliziers. J. Lightwave Technol. 17 (1999) 1602-1616
2. Rasmussen, J. C.: Automatic PMD and Chromatic Dispersion Compensation in High Capacity Transmission. In: 2003 Digest of the LEOS Summer Topical Meetings, (2003) 47-48.
3. Kogelnik, H., Jopson, R. M., Nelson, L.: Polarization-Mode Dispersion, In: Kaminow, I. P., Li, T. (eds): Optical Fiber Telecommunications, IV B. Academic Press, San Diego San Francisco New York Boston London Sydney Tokyo, (2002) 725-861
4. Kim, S.: Schemes for Complete Compensation for Polarization Mode Dispersion up to Second Order. Opt. Lett. 27 (2002) 577-579
5. Kennedy, J., Eberhart, R. C.: Paticle Swarm Optimization. In: Proc. of IEEE International Conference on Neural Networks. Piscataway, NJ, USA, (1995) 1942-1948
6. Laskari, E. C., Parsopoulos, K. E., Vrahatis, M. N.: Particle Swarm Optimization for Minimax Problems," In: Proc. of the 2002 Congress on Evolutionary Computation. Vol.2. (2002) 1576-1581
7. Zhang, X. G., Yu, L., Zheng, Y., Shen, Y., Zhou, G. T., Chen, L., Xi, L. X., Yuan, T. C., Zhang, J. Z., Yang, B. J.: Two-Stage Adaptive PMD Compensation in 40Gb/s OTDM Optical Communication System Using PSO Algorithm. Opt. Quantum Electron. 36 (2004) 1089-1104
8. Kennedy, J., Mendes, R.: Population Structure and Particle Swarm Performance. In: Proc. of the 2002 Congress on Evolutionary Computation. Vol.2. (2002) 1671-1676
9. Eberhart, R. C., Kennedy, J.: A New Optimizer Using Particle Swarm Theory. In: Proc. of the Sixth International Symposium on Micro Machine and Human Science. (1995) 39-43

Prediction of Construction Litigation Outcome Using a Split-Step PSO Algorithm

Kwok-wing Chau

Department of Civil and Structural Engineering, Hong Kong Polytechnic University,
Hunghom, Kowloon, Hong Kong
cekwchau@polyu.edu.hk

Abstract. The nature of construction claims is highly complicated and the cost involved is high. It will be advantageous if the parties to a dispute may know with some certainty how the case would be resolved if it were taken to court. The recent advancements in artificial neural networks may render a cost-effective technique to help to predict the outcome of construction claims, on the basis of characteristics of cases and the corresponding past court decisions. In this paper, a split-step particle swarm optimization (PSO) model is applied to train perceptrons in order to predict the outcome of construction claims in Hong Kong. It combines the advantages of global search capability of PSO algorithm in the first step and the local convergence of back-propagation algorithm in the second step. It is shown that, through a real application case, its performance is much better than the benchmark backward propagation algorithm and the conventional PSO algorithm.

1 Introduction

The nature of construction activities is varying and dynamic, which can be evidenced by the fact that no two sites are exactly the same. Thus the preparation of the construction contract can be recognized as the formulation of risk allocation amongst the involving parties: the client, the contractor, and the engineer. The risks involved include the time of completion, the final cost, the quality of the works, inflation, inclement weather, shortage of materials, shortage of plants, labor problems, unforeseen ground conditions, site instructions, variation orders, client-initiated changes, engineer-initiated changes, errors and omissions in drawings, mistakes in specifications, defects in works, accidents, supplier delivery failure, delay of schedule by subcontractor, poor workmanship, delayed payment, changes in regulations, third-party interference, professional negligence, and so on.

Before the actual construction process, the involving parties will attempt to sort out the conditions for claims and disputes through the contract documents. However, since a project usually involves thousands of separate pieces of work items to be integrated together to constitute a complete functioning structure, the potential for honest misunderstanding is extremely high. The legislation now in force requires that any disputes incurred have to be resolve successively by mediation, arbitration, and the courts [1].

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1101 – 1107, 2006.

By its very nature, the construction industry is prone to litigation since claims are normally affected by a large number of complex and interrelated factors. However, the consequence of any disagreements between the client and the contractor may be far reaching. It may lead to damage to the reputation of both sides, as well as inefficient use of resources and higher costs for both parties through settlement. The litigation process is usually very expensive since it involves specialized and complex issues. Thus, it is the interest of all the involving parties to minimize or even avoid the likelihood of litigation through conscientious management procedure and concerted effort. It is highly desirable for the parties to a dispute to know with some certainty how the case would be resolved if it were taken to court. This would effectively help to significantly reduce the number of disputes that would need to be settled by the much more expensive litigation process.

Recently, soft computing (SC) techniques have been gradually becoming a trend. The characteristics of these data-driven approaches include built-in dynamism, data-error tolerance, no need to have exogenous input and so on. Amongst others, artificial neural networks (ANN), in particular the feed forward back-propagation (BP) perceptrons, have been widely applied in different fields [2-6]. The use of ANN can be a cost-effective technique to help to predict the outcome of construction claims, on the basis of characteristics of cases and the corresponding past court decisions. It can be used to identify the hidden relationships among various interrelated factors and to mimic decisions that were made by the court. However, slow training convergence speed and easy entrapment in a local minimum are inherent drawbacks of the commonly used BP algorithm [7]. Swarm intelligence is another recent SC technique that is developing quickly [8]. These SC techniques have been applied successfully to different areas [9-12].

This paper presents a split-step PSO algorithm which is employed to train multi-layer perceptrons for prediction of the outcome of construction litigation in Hong Kong. It is believed that, by combining the two algorithms, the advantages of global search capability of PSO algorithm in the first step and local convergence of BP algorithm in the second step can be fully utilized to furnish promising results. This paper contributes to the verification of this new algorithm to real prototype application. It can be extended and applied to other areas as well.

2 Split-Step PSO Algorithm

The combination of two different SC techniques could enhance the performance through fully utilization of the strengths of each technique. In this algorithm, the training process is divided into two stages. Initially the perceptron is trained with the PSO algorithm for a predetermined generation number to exploit the global search ability for near-optimal weight matrix. Then, after this stage, the perceptron is trained with the BP algorithm to fine tune the local search. This might be able to avoid the drawback of either entrapment in local minima in BP algorithm or longer time consumption in global search of PSO algorithm.

2.1 PSO Algorithm

When PSO algorithm is initially proposed, it is considered a tool for modeling social behavior and for optimization of difficult numerical solutions [8,13]. This computational intelligence technique is intimately related to evolutionary algorithms and is an optimization paradigm that mimics the ability of human societies to process knowledge [14]. Its principle is based on the assumption that potential solutions will be flown through hyperspace with acceleration towards more optimum solutions. PSO is a populated search method for optimization of continuous nonlinear functions resembling the biological movement in a fish school or bird flock. Each particle adjusts its flying according to the flying experiences of both itself and its companions. During the process, the coordinates in hyperspace associated with its previous best fitness solution and the overall best value attained so far by other particles within the group are kept track and recorded in the memory.

One of the more significant advantages is its relatively simple coding and hence low computational cost. One of the similarities between PSO and a genetic algorithm is the fitness concept and the random population initialization. However, the evolution of generations of a population of these individuals in such a system is by cooperation and competition among the individuals themselves. The population is responding to the quality factors of the previous best individual values and the previous best group values. The allocation of responses between the individual and group values ensures a diversity of response. The principle of stability is adhered to since the population changes its state if and only if the best group value changes. It is adaptive corresponding to the change of the best group value. The capability of stochastic PSO algorithm, in determining the global optimum with high probability and fast convergence rate, has been demonstrated in other cases [13-14].

2.2 Training of Three-Layered Perceptrons

PSO can be readily adopted to train the multi-layer perceptrons as an optimization technique. In the following section, a three-layered preceptron is considered, although the same principle still holds for other number of layers. $W^{[1]}$ and $W^{[2]}$ represent the connection weight matrix between the input layer and the hidden layer, and that between the hidden layer and the output layer, respectively. During training of the preceptron, the i-th particle is denoted by $W_i = \{W^{[1]}, W^{[2]}\}$ whilst the velocity of particle i is denoted by V_i. The position representing the previous best fitness value of any particle is denoted by P_i whilst the best matrix among all the particles in the population is recorded as P_b. Let m and n represent the index of matrix row and column, respectively, the following equation represents the computation of the new velocity of the particle based on its previous velocity and the distances of its current position from the best experiences both in its own and as a group.

$$\begin{aligned} V_i^{[j]}(m,n) = V_i^{[j]}(m,n) + r\alpha[P_i^{[j]}(m,n) - W_i^{[j]}(m,n)] \\ + s\beta[P_b^{[j]}(m,n) - W_i^{[j]}(m,n)] \end{aligned} \tag{1}$$

where j = 1, 2; m = 1, …, M_j; n= 1, …, N_j; M_j and N_j are the row and column sizes of the matrices W, P, and V; r and s are positive constants; α and β are random numbers in the range from 0 to 1. In the context of social behavior, the cognition part $r\alpha[P_i^{[j]}(m,n) - W_i^{[j]}(m,n)]$ denotes the private thinking of the particle itself whilst the social part $s\beta[P_b^{[j]}(m,n) - W_i^{[j]}(m,n)]$ represents the collaboration among the particles as a group. The new position is then determined based on the new velocity as follows:

$$W_i^{[j]} = W_i^{[j]} + V_i^{[j]} \tag{2}$$

The fitness of the i-th particle is determined in term of an output mean squared error of the neural networks as follows:

$$f(W_i) = \frac{1}{S}\sum_{k=1}^{S}\left[\sum_{l=1}^{O}\{t_{kl} - p_{kl}(W_i)\}^2\right] \tag{3}$$

where f is the fitness value, t_{kl} is the target output; p_{kl} is the predicted output based on W_i; S is the number of training set samples; and, O is the number of output neurons.

3 Application to Construction Litigation

In this study, the system is applied to predict the outcome of construction claims in Hong Kong. The existing data from 1991 to 2000 are pre-processed initially and organized case by case in order to correlate the relationship between the dispute characteristics and court decisions. Through a sensitivity analysis, 13 case elements that seem relevant in courts' decisions, which are namely, type of contract, contract value, parties involved, type of plaintiff, type of defendant, resolution technique involved, legal interpretation of contract documents, misrepresentation of site, radical changes in scope, directed changes, constructive changes, liquidated damages involved, and late payment, are identified.

As far as possible, the 13 case elements are expressed in binary format; for example, the input element 'liquidated damages involved' receives a 1 if the claim involves liquidated damages or a 0 if it does not. However, some elements are defined by several alternatives; for example, 'type of contract' could be remeasurement contract, lump sum contract, or design and build contract. These elements with alternative answers are split into separate input elements, one for each alternative. Each alternative is represented in a binary format, such as 1 for remeasurement contract and 0 for the others if the type of contract is not remeasurement. In that case, only one of these input elements will have a 1 value and all the others will have a 0 value. In this way, the 13 elements are converted into an input layer of 30 neurons, all expressed in binary format. The court decisions are also organized in an output layer of 6 neurons expressed in binary format corresponding to the 6 elements: client, contractor, engineer, sub-contractor, supplier, and other third parties. Table 1 shows examples of the input neurons for cases with different types of contract.

Table 1. Examples of the input neurons for cases with different types of contract

Input neuron	Cases		
	Remeasurement	Lump sum	Design and build
Type of contract - remeasurement	1	0	0
Type of contract - lump sum	0	1	0
Type of contract – design and build	0	0	1

In this case, 1105 sets of construction-related cases are employed, of which 550 from years 1991 to 1995 are used for training, 275 from years 1996 to 1997 are used for testing, and 280 from years 1998 to 2000 are used to validate the network results with the observations. In the PSO-based perceptron, the number of population is set to be 40 whilst the maximum and minimum velocity values are 0.25 and -0.25 respectively. In forming the data series for training and validation, a balanced distribution of cases is ensured. In order to determine the best architecture, a sensitivity analysis is undertaken to vary in the number of hidden layers and number of hidden neurons. After a lot of numerical experiments, the final perceptron is determined. Table 2 shows the parameters for the best architecture.

Table 2. Parameters for the best architecture

	Parameter
No. of hidden layer	3
No. of neuron in input layer	30
No. of neuron in hidden layer	15
No. of neuron in output layer	6

Table 3. Comparison of prediction results for various perceptrons

Algorithm	Training		Validation	
	Coefficient of correlation	Prediction rate	Coefficient of correlation	Prediction rate
BP-based	0.956	0.69	0.953	0.67
PSO-based	0.987	0.81	0.984	0.80
Split-step	0.988	0.83	0.985	0.82

4 Analysis and Discussions

In evaluating the performance of the split-step multi-layer ANN, a comparison is made with several commonly used existing methods, i.e., the benchmarking standard

BP-based network and a PSO-based network. A fair and common initial ground is ensured for comparison purpose as far as possible. The training process of the BP-based perceptron commences from the best initial population of the corresponding PSO-based perceptron or split-step network. Table 3 shows comparisons of the results of network for various perceptrons. It can be observed that the split-step algorithm performs the best in terms of prediction accuracy. It is noted that testing cases of the split-step PSO-based network are able to give a successful prediction rate higher than 80%, which is much higher than by pure chance.

Table 4 shows the steady-state fitness evaluation times during training for various perceptrons. The fitness evaluation time here for the PSO-based perceptron is equal to the product of the population with the number of generations. It can be observed that the split-step perceptron exhibits much faster convergence than those by the BP-based perceptron and the PSO-based network. It is, of course, recognized that there are limitations in the assumptions used in this study. Other factors that may have certain bearing such as cultural, psychological, social, environmental, and political factors have not been considered here.

Table 4. Steady-state fitness evaluation times during training for various perceptrons

Algorithm	Steady-state fitness valuation time
BP-based	22,400
PSO-based	8,300
Split-step	7,900

5 Conclusions

This paper presents the application of a perceptron based on a split-step PSO algorithm for prediction of outcomes of construction litigation on the basis of the characteristics of the individual dispute and the corresponding past court decisions. It is believed that, if the involving parties to a construction dispute become aware with some certainty how the case would be resolved if it were taken to court, the number of disputes could be reduced significantly. It is shown that the split-step PSO-based perceptron performs much better than the other commonly used optimization techniques in prediction of outcomes of construction litigation. The rate of prediction for the network finally adopted in this study is higher than 80%, which is much higher than pure chance. It can be used as a good prediction tool for the parties in dispute.

References

1. Chau, K.W.: Resolving Construction Disputes by Mediation: Hong Kong Experience. Journal of Management in Engineering, ASCE **8(4)** (1992) 384-393
2. Chau, K.W.: A Review on the Integration of Artificial Intelligence into Coastal Modeling. Journal of Environmental Management **80(1)** (2006) 47-57
3. Chau, K.W., Cheng, C.T.: Real-Time Prediction of Water Stage with Artificial Neural Network Approach. Lecture Notes in Artificial Intelligence **2557** (2002) 715-715

4. Cheng, C.T., Chau, K.W., Sun, Y.G., Lin, J.Y.: Long-term Prediction of Discharges in Manwan Reservoir using Artificial Neural Network Models. Lecture Notes in Computer Science **3498** (2005) 1040-1045
5. Cheng, C.T., Lin, J.Y., Sun, Y.G., Chau, K.W.: Long-Term Prediction of Discharges in Manwan Hydropower using Adaptive-Network-Based Fuzzy Inference Systems Models. Lecture Notes in Computer Science **3612** (2005) 1152-1161
6. Wu, C.L. Chau, K.W.: Evaluation of Several Algorithms in Forecasting Flood. Lecture Notes in Artificial Intelligence **4031** (2006) 111-116
7. Rumelhart, D.E., Widrow, B., Lehr, M.A.: The Basic Ideas in Neural Networks. Communications of the ACM **37(3)** (1994) 87-92
8. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. Proceedings of the 1995 IEEE International Conference on Neural Networks. Perth (1995) 1942-1948
9. Chau, K.W.: River Stage Forecasting with Particle Swarm Optimization. Lecture Notes in Artificial Intelligence **3029** (2004) 1166-1173
10. Chau, K.W.: Rainfall-Runoff Correlation with Particle Swarm Optimization Algorithm. Lecture Notes in Computer Science **3174** (2004) 970-975
11. Chau, K.W.: Predicting Construction Litigation Outcome using Particle Swarm Optimization. Lecture Notes in Artificial Intelligence **3533** (2005) 571-578
12. Arditi, D., Oksay, F.E., Tokdemir, O.B.: Predicting the Outcome of Construction Litigation Using Neural Networks. Computer-Aided Civil and Infrastructure Engineering **13(2)** (1998) 75-81
13. Kennedy, J.: The Particle Swarm: Social Adaptation of Knowledge. Proceedings of the 1997 International Conference on Evolutionary Computation. Indianapolis (1997) 303-308
14. Clerc, M., Kennedy, J.: The Particle Swarm—Explosion, Stability, and Convergence in a Multidimensional Complex Space. IEEE Transactions on Evolutionary Computation **6(1)** (2002) 58-73

Solving Multiprocessor Real-Time System Scheduling with Enhanced Competitive Scheme

Ruey-Maw Chen[1], Shih-Tang Lo[2], and Yueh-Min Huang[2]

[1] Department of Computer Science and Information Engineering, National Chin-yi Institute of Technology, Taichung 411, Taiwan, ROC
[2] Department of Engineering Science, National Cheng-Kung University, Tainan 701, Taiwan, ROC
raymond@chinyi.ncit.edu.tw, edwardlo@mail.ksu.edu.tw, huang@mail.ncku.edu.tw

Abstract. A new method based on Hopfield Neural Networks (HNN) for solving real-time scheduling problem is adopted in this study. Neural network using competitive learning rule provides a highly effective method and deriving a sound solution for scheduling problem. Moreover, competitive scheme reduces network complexity. However, competitive scheme is a *1-out-of-N* confine rule and applicable for limited scheduling problems. Restated, the processor may not be full utilization for scheduling problems. To facilitate the non-fully utilized problem, extra neurons are introduced to the Competitive Hopfield Neural Network (CHNN). Slack neurons are imposed on CHNN with respected to pseudo processes. Simulation results reveal that the competitive neural network imposed on the proposed energy function with slack neurons integrated ensures an appropriate approach of solving both full and non-full utilization multiprocessor real-time system scheduling problems.

Keyword: Hopfield neural network, Scheduling, Slack neuron, Competitive learning.

1 Introduction

A real-time job scheduling problem is a timing constraint problem. Many approaches for solving the optimization problems are proposed. Artificial Neural Networks (ANN) has been widely used in many applications like operations research, production planning, image processing, identification and control, etc. A competitive neural network provides a highly effective means of attaining a sound solution and of reducing the network complexity.

In general, job scheduling problems are seen as involving to execute a set of jobs satisfying a given type of constraints and optimizing a given criterion. Jobs are assigned timing constraints like ready time and deadline, and processing time [1]. Liu and Leyland was the pioneering paper about real time scheduling algorithms for mono-job or scheduling of independent and periodic tasks [2]. Willems and Rooda translated the job-shop scheduling problem onto a linear programming format, and then mapped it into an appropriate neural network structure to obtain a solution [3].

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1108 – 1117, 2006.

Furthermore Foo and Takefuji et al. adopted integer linear programming neural networks to solve the scheduling problem by minimizing the total starting times of all jobs by a precedence constraint [4]. Yan and Chang developed a neural network algorithm derived from linear programming, in which preemptive jobs are scheduled according to their priorities and deadline [5]. Silva et al. explored the multi-process real-time scheduling with a HNN [6]. Above investigations concentrating on the preemptive jobs executed on multiple processors with job transfer by a neural network. Moreover, Hanada and Ohnishi [7] presented a parallel algorithm based on a neural network for task scheduling problems by permitting task transfer among processors. A classical local search heuristic algorithm was embedded into the TSP by Park [8]. In real-time applications, failure to meet timing constraints of system may lead to a hazardous situation. A modified neural network with slack neurons is constructed to solve the scheduling problems. In the HNN [9], the state input information from a community of neurons is received to decide neuron output state information. These neurons apply this information to cooperatively move the network to achieve convergence. The energy function used in the HNN is an appropriate Lyapunov function. Dixon et al. applied the HNN with mean field annealing to solve the shortest path problem in a communication network [10]. In our previous work also solved a multi-constraint schedule problem for a multi-processor or system by the HNN [11].

A CHNN applies a competitive learning mechanism to update the neuron states in the HNN. A competitive learning rule provides a highly effective method of attaining a sound solution and is capable of simplifying the network complexity. CHNN has been applied in image clustering processes and specific image segmentation [12][13]. The winner-take-all rule employed by the competitive learning mechanism ensures that only one job is executed on a dedicated machine at a certain time, enforcing the *1-out-of-N* constraint to be held. The maximum output value neuron of the set of neurons is activated. The monotonic of the maximum neuron follows the fact that it is equivalent to a McCulloch and Pitts neuron with a dynamic threshold [14]. A series of studies has been done using HNN and mean field annealing (MFA) techniques are utilized to multi-processor scheduling problem [15][16]. Cardeira and Mammeri investigated the multi-process and real-time scheduling to meet deadline requirements by applying the *k-out-of-N* rule, which extends slack neurons to a neural network to agree with the inequality constraints. They extended the methodology to handle real-time scheduling with precedence constraints [17][18]. Tagliarini et al. demonstrated a weapon-to-target approach for a resource allocation task problem [19]. A slack neuron is associated with each weapon. The slack neuron activated represents the hypotheses that the associated weapon is not fired. In real-time scheduling problem, due to the capacity constraints or availability of resources, the processors may not reach full utilization. This work explores the real-time job scheduling problem on a non-fully utilized (incomplete usage) system including timing constraints. Extra slack neurons are added on to the networks to meet fully utilized conditions.

The rest of this paper is organized as follows. Section 2 derives the corresponding energy function of scheduling problem according to the intrinsic constraints. Section 3 reviews the competitive algorithm with slack neuron and translates the derived energy functions to the proposed algorithm. The simulation examples and experimental results are presented in Section 4. The conclusions showed in Section 5.

2 Energy Function of the Scheduling Problem

The scheduling problem in this work is defined as follows. First, a job can be segmented with no job precedence relation, and the execution of each segment is preemptive. Second, job migration is not allowed between processors. Third, each job's execution time and deadline and no setup time requirement.

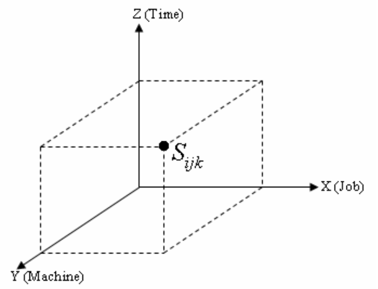


Fig. 1. 3-D Hopfield neural network

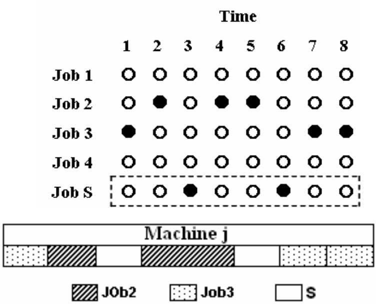


Fig. 2. Neural network with slack neurons and corresponding Gantt chart expression

The state variables S_{ijk} are displayed in Fig. 1. The “x” axis denotes the “job” variable, with i representing a specific job with a range from 1 to N+1, where N is the total number of jobs to be scheduled. The $(N+1)^{th}$ job is a pseudo-job. It is a supplementary job to fulfill *1-out-of-N* rule for each column on a dedicated processor as shown in Fig. 2. The additional neurons are analogous to slack variables that are sometimes adopted to solve optimization problems in operation research, and are therefore called “slack neurons”. Herein, slack neurons are neurons in representing the pseudo-job. One processor processes a pseudo-job, indicating that the processor is doing nothing at this time as displayed at time 3 and 6 in Fig. 2. The “y” axis represents the “processor” variable, and the term j on the axis represents a dedicated processor from 1 to M, where M denotes the total number of processors. Finally, the “z” axis denotes the “time” variable, with k representing a specific time which should be less than or equal to T, where T is the job deadline. Thus, a state variable S_{ijk} is defined as representing whether or not job i is executed on processor j at a certain time k. The activated neuron $S_{ijk}=1$ denotes that the job i is run on processor j at time k; otherwise, $S_{ijk}=0$.

There are five energy terms of energy function to represent the problem. The first term is to ensure that processor j can only run one job at a certain time k. If job i is processed on processor j at time k (S_{ijk}=1), there is no other job i' can be processed on same processor at the same time. This energy term is defined as

$$\sum_{i=1}^{N+1}\sum_{j=1}^{M}\sum_{k=1}^{T}\sum_{\substack{i'=1\\ i'\neq i}}^{N+1} S_{ijk}S_{i'jk}\ . \qquad (1)$$

where *N, M, T, i, j, k, i',* and S_{ijk} are as defined above, the rest of this study employs the same notations. This term has a minimum value of zero when it meets this constraint, which arises when S_{ijk}=0 or $S_{i'jk}$=0. The second term confines job migration, indicating that job *i* runs on processor *j* or *j'*. If a job is assigned on a dedicated processor, then all of its segments must be executed on the same processor, which is the non-migration constraint. However, the $(N+1)^{th}$ job is an exception which can be processed on different processors. Accordingly, the energy term is defined as follows:

$$\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{T}\sum_{\substack{j'=1\\ j'\neq j}}^{M}\sum_{k'=1}^{T} S_{ijk}S_{ij'k'}\,. \tag{2}$$

This term also has a minimum value of zero when S_{ijk} or $S_{ij'k'}$ is zero. The third energy term is defined to meet the process time constraint as

$$\sum_{i=1}^{N+1}\left(\sum_{j=1}^{M}\sum_{k=1}^{T} S_{ijk}-P_i\right)^2. \tag{3}$$

where P_i is the process time of job *i*. For a feasible solution, the total processing time of job *i* is no more or less than P_i, such that $\sum\sum S_{ijk}=P_i$, Eq. (3) becomes zero. The processing time of the pseudo-job (the $N+1^{th}$ job) is defined as the total available time for all processors subtracts the total processing time required by all N jobs. Moreover, the third constraint energy term that to ensure no two or more job being executed on a specific processor at a certain time when using the *1-out-of-N* rule, is introduced as

$$\sum_{j=1}^{M}\sum_{k=1}^{T}\left(\sum_{i=1}^{N+1} S_{ijk}-1\right)^2. \tag{4}$$

Therefore, this energy term should also reach a minimum value of zero which satisfy *1-out-of-N* rule. The following energy term is defined to meet the deadline requirement of each job *i*:

$$\sum_{i=1}^{N+1}\sum_{j=1}^{M}\sum_{k=1}^{T} S_{ijk}G_{ijk}^2H(G_{ijk})\,.$$
$$H(G_{ijk})=\begin{cases}1 & if\ G_{ijk}>0\\ 0 & if\ G_{ijk}\le 0\end{cases},G_{ijk}=k-d_i\,. \tag{5}$$

where d_i denotes the deadline of job *i* and $H(G_{ijk})$ is the unit step function. Similarly, the maximum time limit is set to the deadline of the pseudo-job. The energy term will exceed zero when a job is allocated at time *k*, the run time *k* is greater than d_i, i.e., when S_{ijk}=1, k-d_i>0, then $H(G_{ijk})$>0. The energy value grows exponentially with the associated time lag between d_i and *k*, given by k-d_i. Conversely, this energy term has a value of zero if S_{ijk}=1 and k-$d_i \leq 0$, which time job *i* is processed no more than d_i. Accordingly, the energy function with all constraints can be induced as shown in Eq. (6).

$$E = \frac{C_1}{2}\sum_{i=1}^{N+1}\sum_{j=1}^{M}\sum_{k=1}^{T}\sum_{\substack{i'=1\\ ,i'\neq i}}^{N+1} S_{ijk}S_{i'jk} + \frac{C_2}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{T}\sum_{\substack{j'=1\\ ,j'\neq j}}^{M}\sum_{k'=1}^{T} S_{ijk}S_{ij'k'}$$
$$+\frac{C_3}{2}\sum_{i=1}^{N+1}(\sum_{j=1}^{M}\sum_{k=1}^{T} S_{ijk} - P_i)^2 + \frac{C_4}{2}\sum_{j=1}^{M}\sum_{k=1}^{T}(\sum_{i=1}^{N+1} S_{ijk} - 1)^2 \tag{6}$$
$$+\frac{C_5}{2}\sum_{i=1}^{N+1}\sum_{j=1}^{M}\sum_{k=1}^{T} S_{ijk}G^2{}_{ijk}H(G_{ijk}) .$$

C_1, C_2, C_3, C_4, and C_5 are weighting factors, they are assumed to be positive constants. Based on the discussion above, the derived energy function has a minimum value of zero when all constraints are met. Equation (6) can be proved to be an appropriate Lyapunov function for the system.

3 Competitive Algorithm

In this section, the defined energy functions are transformed onto the CHNN. In [20] and [21], a circuit composed of simple analog amplifiers that implements this type of neural networks was proposed. Based on dynamic system theory, the Lyapunov function [20] shown in Eq. (7) has verified the existence of stable states of the network system. The energy function representing the scheduling problem must be in the same format as the Liapunov function and expanded to a three-dimensional model as below

$$E = -\frac{1}{2}\sum_{x}\sum_{y}\sum_{z}\sum_{i}\sum_{j}\sum_{k} S_{xyz}W_{xyzijk}S_{ijk} + \sum_{i}\sum_{j}\sum_{k}\theta_{ijk}S_{ijk} . \tag{7}$$

S_{xyz} and S_{ijk} denote the neuron states, W_{xyzijk} represent the synaptic weight among neurons, and θ_{ijk} denotes the threshold value representing the bias input of the neuron. The conventional HNN uses the deterministic rule to update the neuron state. The deterministic rule is displayed in Eq. (8).

$$S_{ijk}^{n+1} = \begin{cases} 1, & if \quad Net_{ijk} > 0 \\ S_{ijk}^{n}, & if \quad Net_{ijk} = 0 . \\ 0, & if \quad Net_{ijk} < 0 \end{cases} \tag{8}$$

Meanwhile, Net_{ijk} represents the net value of the neuron *(i, j, k)* obtained by the Eq. (9) as follows:

$$Net_{ijk} = -\frac{\partial E}{\partial S_{ijk}} = \sum_{x}\sum_{y}\sum_{z} W_{xyzijk}S_{xyz} - \theta_{ijk} . \tag{9}$$

Instead of applying conventional deterministic rules to update the neuron states, this study uses competition rule to decide the winning neuron among the set of neurons, i.e., the active neuron. As discussed previously, a HNN applying a winner-take-all learning mechanism is called a competitive Hopfield neural network, CHNN. The competitive rule is adopted to decide “exactly one neuron among N neurons” and can be regarded as a *1-out-of-N* confine rule. Hence, the number of activated neurons during each time unit is exactly the number of processors.

Since one processor can only execute one job at certain time in a subject scheduling problem. Thus, the C1 and C4 energy terms are omitted from Eq. (6). Restated, the first C1 and the fourth C4 energy terms are handled implicitly in *1-out-of-N* competitive rule. The resulting simplified energy function is as follows:

$$E=\frac{C2}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{T}\sum_{\substack{j'=1\\ j'\neq j}}^{M}\sum_{k'=1}^{T}S_{ijk}S_{ij'k'}+\frac{C3}{2}\sum_{i=1}^{N+1}(\sum_{j=1}^{M}\sum_{k=1}^{T}S_{ijk}-P_i)^2$$
$$+\frac{C5}{2}\sum_{i=1}^{N+1}\sum_{j=1}^{M}\sum_{k=1}^{T}S_{ijk}G_{ijk}^2H(G_{ijk})\,. \tag{10}$$

This simplified energy function must be an appropriate Lyapunov function.

The synaptic interconnection strength W_{xyzijk} and the bias input θ_{ijk} can be obtained by comparing Eq. (10) with Eq. (7) where

$$W_{xyzijk}=-C2*\delta(x,i)*(1-\delta(y,j))-C3*\delta(x,i)\,. \tag{11}$$

and

$$\theta_{xyz}=-C3P_i+\frac{C5}{2}*G^2*H(G)\,. \tag{12}$$

respectively, where

$$\delta(a,b)=\begin{cases}1 & if\ a=b\\ 0 & if\ a\neq b\end{cases}.\ \text{is the } \textit{Kronecker delta} \text{ function.} \tag{13}$$

The CHNN imposed a competitive winner-take-all rule to update the neuron states. Neurons on the *same column* of a dedicated processor at a given time compete with one another to determine the winning neuron. The neuron that receives the highest net value is the winning neuron. Accordingly, the output of the winner neuron is set to 1, and the output states of all the other neurons on the same column are set to 0. For example, there are four jobs to be processed on two machines (processors) as displayed in Fig. 2. If jobs 2 and 3 are assigned to machine 1, then the neuron activated ($S_{ijk}=1$) is shown by a solid node. Therefore, the final neural network state has exactly one activated neuron at a time for each machine. Restated, the neural state determination is regarded as a *1-out-of-N+1* rule. The winner-take-all update rule of the neuron for the i^{th} column is illustrated as follows:

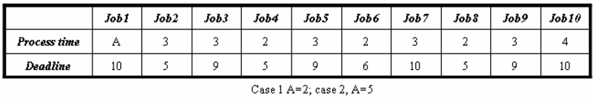
$$S_{xjk}=\begin{cases}1 & if\ \ Net_{xjk}=\underset{i=1\sim N+1}{Max}\,Net_{ijk}\\ 0 & otherwise\end{cases}. \tag{14}$$

where Net_{xjk} denotes the maximum total neuron input which is equivalent to the dynamic threshold on a McCulloch and Pitts neuron [14].

4 Experimental simulations

The simulations involve different sets of scheduling problems with timing constraints and use various set of weighting factors. The C2, C3, and C5 weighting factors used in the following simulation results were set to 1.35, 0.55 and 1.3 respectively. Table 1 shows the timing constraints of simulation cases for 10 jobs on three processors (machines). Case 1 is the non-full utilization situation. Case 2 is the full utilization example. Case 2 is the same simulation example as in [16].

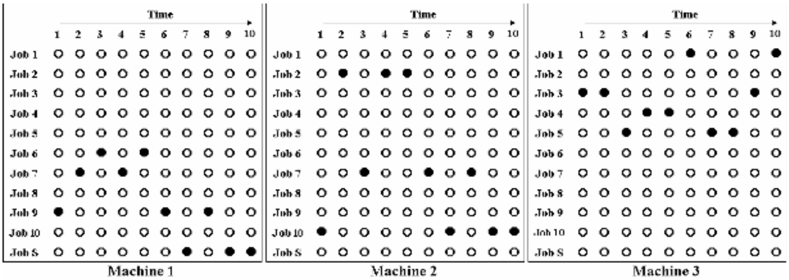
Table 1. Timing matrix of simulations



	Job1	*Job2*	*Job3*	*Job4*	*Job5*	*Job6*	*Job7*	*Job8*	*Job9*	*Job10*
Process time	A	3	3	2	3	2	3	2	3	4
Deadline	10	5	9	5	9	6	10	5	9	10

Case 1 A=2; case 2, A=5

The simulation results were displayed with neural states to graphically represent the job schedules. The neural states graph can be transferred into a Gantt chart expression as shown in Fig. 2. In our previous work [16], this is a full utilization problem. In this investigation, the proposed method solves both non-full utilization and full utilization real-time scheduling problems. Figures 3 and 4 illustrate the resulting schedules of case 1 and case 2 by the proposed algorithm. The job S as displayed in figures indicates slack neurons. Restated, the processor does nothing at that time while the slack neuron is activated. To minimize the completion time of a processor, the jobs behind the slack neurons can shift forward if there are no other constraints violated.



	1	2	3	4	5	6	7	8	9	10
Job 1	○	○	○	○	○	○	○	○	○	○
Job 2	○	○	○	○	○	○	○	○	○	○
Job 3	○	○	○	○	○	○	○	○	○	○
Job 4	○	○	○	○	○	○	○	○	○	○
Job 5	○	○	○	○	○	○	○	○	○	○
Job 6	○	○	●	○	●	○	○	○	○	○
Job 7	○	●	○	●	○	○	○	○	○	○
Job 8	○	○	○	○	○	○	○	○	○	○
Job 9	●	○	○	○	○	●	○	●	○	○
Job 10	○	○	○	○	○	○	○	○	○	○
Job S	○	○	○	○	○	○	●	○	●	●

Machine 1

	1	2	3	4	5	6	7	8	9	10
Job 1	○	○	○	○	○	○	○	○	○	○
Job 2	○	●	○	●	●	○	○	○	○	○
Job 3	○	○	○	○	○	○	○	○	○	○
Job 4	○	○	○	○	○	○	○	○	○	○
Job 5	○	○	○	○	○	○	○	○	○	○
Job 6	○	○	○	○	○	○	○	○	○	○
Job 7	○	○	●	○	○	●	○	●	○	○
Job 8	○	○	○	○	○	○	○	○	○	○
Job 9	○	○	○	○	○	○	○	○	○	○
Job 10	●	○	○	○	○	○	●	○	●	●
Job S	○	○	○	○	○	○	○	○	○	○

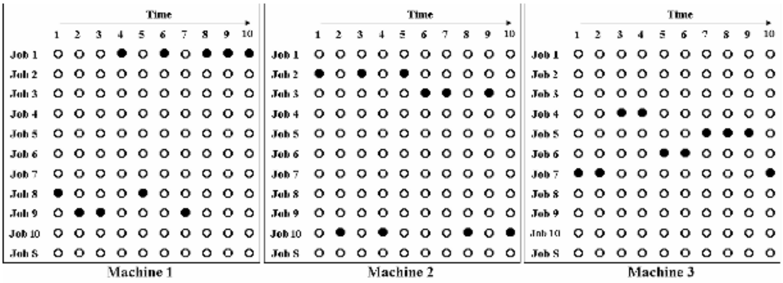
Machine 2

	1	2	3	4	5	6	7	8	9	10
Job 1	○	○	○	○	○	●	○	○	○	●
Job 2	○	○	○	○	○	○	○	○	○	○
Job 3	●	●	○	○	○	○	○	○	●	○
Job 4	○	○	○	●	●	○	○	○	○	○
Job 5	○	○	●	○	○	○	●	●	○	○
Job 6	○	○	○	○	○	○	○	○	○	○
Job 7	○	○	○	○	○	○	○	○	○	○
Job 8	○	○	○	○	○	○	○	○	○	○
Job 9	○	○	○	○	○	○	○	○	○	○
Job 10	○	○	○	○	○	○	○	○	○	○
Job S	○	○	○	○	○	○	○	○	○	○

Machine 3

Fig. 3. Job assignment for case 1

Moreover, different initial neuron states are simulated to better understand the response of the neural network to the scheduling problem. Figure 5 is the simulation result of case 1 under different initial neuron states. Figure 6 displays the resulting schedules correlating with different initial neuron states for case 2. Different initial neuron states generate different feasible solutions. Notably, to guarantee convergence to a minimum, the neuron state update was performed sequentially with complete neuron update each time in the simulation.



Time

	1	2	3	4	5	6	7	8	9	10
Job 1	○	○	○	●	○	●	○	●	●	●
Job 2	○	○	○	○	○	○	○	○	○	○
Job 3	○	○	○	○	○	○	○	○	○	○
Job 4	○	○	○	○	○	○	○	○	○	○
Job 5	○	○	○	○	○	○	○	○	○	○
Job 6	○	○	○	○	○	○	○	○	○	○
Job 7	○	○	○	○	○	○	○	○	○	○
Job 8	●	○	○	○	●	○	○	○	○	○
Job 9	○	●	●	○	○	○	●	○	○	○
Job 10	○	○	○	○	○	○	○	○	○	○
Job S	○	○	○	○	○	○	○	○	○	○

Machine 1

Time

	1	2	3	4	5	6	7	8	9	10
Job 1	○	○	○	○	○	○	○	○	○	○
Job 2	●	○	●	○	●	○	○	○	○	○
Job 3	○	○	○	○	○	●	●	○	●	○
Job 4	○	○	○	○	○	○	○	○	○	○
Job 5	○	○	○	○	○	○	○	○	○	○
Job 6	○	○	○	○	○	○	○	○	○	○
Job 7	○	○	○	○	○	○	○	○	○	○
Job 8	○	○	○	○	○	○	○	○	○	○
Job 9	○	○	○	○	○	○	○	○	○	○
Job 10	○	●	○	●	○	○	○	●	○	●
Job S	○	○	○	○	○	○	○	○	○	○

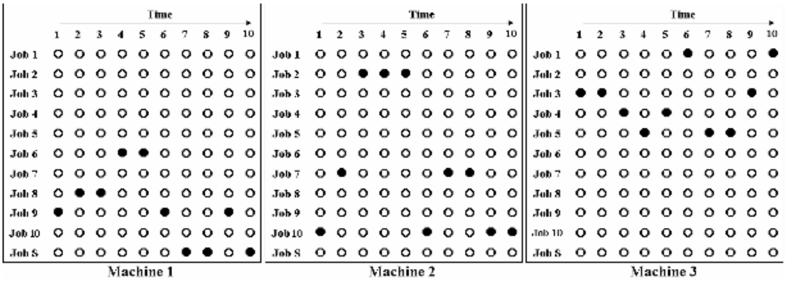
Machine 2

Time

	1	2	3	4	5	6	7	8	9	10
Job 1	○	○	○	○	○	○	○	○	○	○
Job 2	○	○	○	○	○	○	○	○	○	○
Job 3	○	○	○	○	○	○	○	○	○	○
Job 4	○	○	●	●	○	○	○	○	○	○
Job 5	○	○	○	○	○	○	●	●	●	○
Job 6	○	○	○	○	●	●	○	○	○	○
Job 7	●	●	○	○	○	○	○	○	○	●
Job 8	○	○	○	○	○	○	○	○	○	○
Job 9	○	○	○	○	○	○	○	○	○	○
Job 10	○	○	○	○	○	○	○	○	○	○
Job S	○	○	○	○	○	○	○	○	○	○

Machine 3

Fig. 4. Job assignment for case 2



Time

	1	2	3	4	5	6	7	8	9	10
Job 1	○	○	○	○	○	○	○	○	○	○
Job 2	○	○	○	○	○	○	○	○	○	○
Job 3	○	○	○	○	○	○	○	○	○	○
Job 4	○	○	○	○	○	○	○	○	○	○
Job 5	○	○	○	○	○	○	○	○	○	○
Job 6	○	○	○	●	●	○	○	○	○	○
Job 7	○	○	○	○	○	○	○	○	○	○
Job 8	○	●	●	○	○	○	○	○	○	○
Job 9	●	○	○	○	○	●	○	○	●	○
Job 10	○	○	○	○	○	○	○	○	○	○
Job S	○	○	○	○	○	○	●	●	○	●

Machine 1

Time

	1	2	3	4	5	6	7	8	9	10
Job 1	○	○	○	○	○	○	○	○	○	○
Job 2	○	○	●	●	●	○	○	○	○	○
Job 3	○	○	○	○	○	○	○	○	○	○
Job 4	○	○	○	○	○	○	○	○	○	○
Job 5	○	○	○	○	○	○	○	○	○	○
Job 6	○	○	○	○	○	○	○	○	○	○
Job 7	○	●	○	○	○	○	●	●	○	○
Job 8	○	○	○	○	○	○	○	○	○	○
Job 9	○	○	○	○	○	○	○	○	○	○
Job 10	●	○	○	○	○	●	○	○	●	●
Job S	○	○	○	○	○	○	○	○	○	○

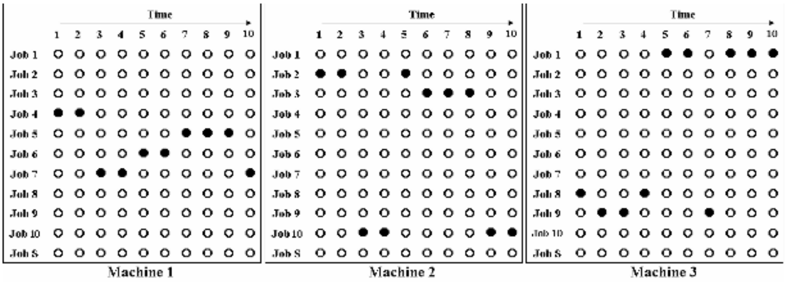
Machine 2

Time

	1	2	3	4	5	6	7	8	9	10
Job 1	○	○	○	○	○	●	○	○	○	●
Job 2	○	○	○	○	○	○	○	○	○	○
Job 3	●	●	○	○	○	○	○	○	●	○
Job 4	○	○	●	○	●	○	○	○	○	○
Job 5	○	○	○	●	○	○	●	●	○	○
Job 6	○	○	○	○	○	○	○	○	○	○
Job 7	○	○	○	○	○	○	○	○	○	○
Job 8	○	○	○	○	○	○	○	○	○	○
Job 9	○	○	○	○	○	○	○	○	○	○
Job 10	○	○	○	○	○	○	○	○	○	○
Job S	○	○	○	○	○	○	○	○	○	○

Machine 3

Fig. 5. Job assignment for case 1 with different initial states



Time

	1	2	3	4	5	6	7	8	9	10
Job 1	○	○	○	○	○	○	○	○	○	○
Job 2	○	○	○	○	○	○	○	○	○	○
Job 3	○	○	○	○	○	○	○	○	○	○
Job 4	●	●	○	○	○	○	○	○	○	○
Job 5	○	○	○	○	○	○	●	●	●	○
Job 6	○	○	○	○	●	●	○	○	○	○
Job 7	○	○	●	●	○	○	○	○	○	●
Job 8	○	○	○	○	○	○	○	○	○	○
Job 9	○	○	○	○	○	○	○	○	○	○
Job 10	○	○	○	○	○	○	○	○	○	○
Job S	○	○	○	○	○	○	○	○	○	○

Machine 1

Time

	1	2	3	4	5	6	7	8	9	10
Job 1	○	○	○	○	○	○	○	○	○	○
Job 2	●	●	○	○	●	○	○	○	○	○
Job 3	○	○	○	○	○	●	●	●	○	○
Job 4	○	○	○	○	○	○	○	○	○	○
Job 5	○	○	○	○	○	○	○	○	○	○
Job 6	○	○	○	○	○	○	○	○	○	○
Job 7	○	○	○	○	○	○	○	○	○	○
Job 8	○	○	○	○	○	○	○	○	○	○
Job 9	○	○	○	○	○	○	○	○	○	○
Job 10	○	○	●	●	○	○	○	○	●	●
Job S	○	○	○	○	○	○	○	○	○	○

Machine 2

Time

	1	2	3	4	5	6	7	8	9	10
Job 1	○	○	○	○	●	●	○	●	●	●
Job 2	○	○	○	○	○	○	○	○	○	○
Job 3	○	○	○	○	○	○	○	○	○	○
Job 4	○	○	○	○	○	○	○	○	○	○
Job 5	○	○	○	○	○	○	○	○	○	○
Job 6	○	○	○	○	○	○	○	○	○	○
Job 7	○	○	○	○	○	○	○	○	○	○
Job 8	●	○	○	●	○	○	○	○	○	○
Job 9	○	●	●	○	○	○	●	○	○	○
Job 10	○	○	○	○	○	○	○	○	○	○
Job S	○	○	○	○	○	○	○	○	○	○

Machine 3

Fig. 6. Job assignment for case 2 with different initial states

This study proposed an approach for solving timing constraint problems with full or not full machine usage problems or with different initial states of neurons. From these simulations, each job has a process time and deadline which were given in advance. The proposed method can solve the real-time job scheduling problem by addressing the problem constraint.

5 Conclusions

The competitive mechanism eliminated the constraint terms in the energy function, simplifying the network by reducing the interconnections among neurons [12], and this is shown in Eq. (10). Hence, the competitive scheme can help overcome the scaling problem.

This investigation illustrated an approach to map the problem constraint into the energy function of the competitive neural network containing slack neurons which were involved so as to resolve the timing constraints schedule problem for both non-full utilization and fully-utilization real-time systems. The proposed competitive scheme with slack neurons is applicable in solving real-time job scheduling problems, even in fully or non-fully utilized scheduling problems. Convergence is initially state dependent, as displayed in Fig. 5 and Fig. 6. Distributing the initial states randomly can generally produce feasible schedules for the investigated scheduling problem. The energy evolution may encounter oscillation behavior during network update. The entailed synaptic weight matrix in Eq. (11) has a symmetric (i.e. $W_{xyzijk} = W_{ijkxyz}$,) property, but nevertheless has a self-feedback interconnection, indicating that $W_{xyzijk}\neq 0$. Therefore, the network may oscillate when it is updated [20]. Consequently, a solution is not guaranteed, causing an inevitable oscillation procedure.

Various sets of weighting factors were investigated. C2 and C3 were tightly coupled since they dominate the synaptic of the network. Different sets of weighting factors may produce different neural network revolutions. However, the reduction of energy terms in this work also assisted in easing this annoying affair.

An important feature of a scheduling algorithm is its efficiency or performance, i.e., how its execution time grows with the problem size. The parameter most relevant to the time a neural network takes to find a solution is the number of iterations needed to converge to a solution. According to the simulation results, the proposed algorithm required an average of 5 ~ 20 epochs to converge. Consequently, this algorithm resulted in a $O((N+1)^2 \times M^2 \times T^2)$ upper bound complexity. Restated, the execution time was proportional to $O(N^2 \times M^2 \times T^2)$ for each epoch. Accordingly, finding the solution for a very large-scale system (very large N and/or very large M) is a drawback of the proposed model. Future works should examine how to reduce the complexities of solving the scheduling problems such as reducing the number of neurons and the interconnections among neurons.

The energy function proposed herein works efficiently and can be applied to similar cases of investigated scheduling problems. The competitive scheme combined with slack neurons suggests that the way to apply this kind of scheduling has inequality constraints. This work concentrated mainly on solving job scheduling without ready time consideration or resource constraints. For more practical implementations, different and more complicated scheduling problems can be further investigated in future researches by applying the proposed algorithm.

References

1. Cardeira, C. & Mammeri, Z. (1996). Neural network versus max-flow algorithms for multi-processor real-time scheduling, Real-Time Systems, Proceedings of the Eighth Euromicro Workshop ,Page(s):175 – 180
2. Liu, C., & Layland, J.(1973). Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment, Journal of the ACM, 20 (l), pp. 46-61.
3. Willems, T. M. & Rooda, J. E. (1994). Neural Networks for Job-shop Scheduling, Control Eng. Practice, 2(1), pp. 31-39.

4. Foo, Y.P.S., & Takefuji, T. (1998). Integer linear programming neural networks for job-shop scheduling. In: IEEE Int. Conf. on Neural Networks, vol. 2, pp. 341-348.
5. Zhang, C.S., Yan, P.F., & Chang, T. (1991). Solving Job-Shop Scheduling Problem with Priority Using Neural Network. In: IEEE Int. Conf. on Neural Networks, pp. 1361-1366.
6. Silva, M.P.,Cardeira C., & Mammeri Z. (1997). Solving real-time scheduling problems with Hopfield-type neural networks. In: EUROMICRO 97 'New Frontiers of Information Technology, Proceedings of the 23rd EUROMICRO Conference, pp. 671 –678.
7. Hanada, A, & Ohnishi, K. (1993). Near optimal jobshop scheduling using neural network parallel computing. In: Int. Conf. on Proc.Industrial Electronics, Control, and Instrumentation, vol. 1, pp. 315-320.
8. Park, J.G., Park, J.M., Kim, D.S., Lee, C.H., Suh, S.W., & Han, M.S. (1994). Dynamic neural network with heuristic. In: IEEE Int. Conf. on Neural Networks, vol. 7, pp. 4650-4654.
9. Hopfield, J.J.,& Tank, D.W.(1985). Neural computation of decision in optimization problems. Biological Cybernetics , 52, 141-152.
10. Dixon, M.W., Cole, G.R., & Bellgard, M.I. (1995). Using the Hopfield Model with Mean-Field Annealing to Solve the Routing Problem in a Communication Network. In: Int. Conf. on Neural Networks, vol. 5, pp. 2652-2657.
11. Huang, Y.M., & Chen, R.M.(1999). Scheduling multiprocessor job with resource and timing constraints using neural network. IEEE Trans. on System, Man and Cybernetics, part B, 29(4): 490-502.
12. Uchiyama, T.,& Arbib, M,A.(1994) Color Image Segmentation Using Competitive Learning. IEEE Trans. Pattern Analysis Machine Intelligence.,16(12): 1197-1206.
13. Chung P.C., Tsai C.T., Chen E.L., & Sun YN. (1994). Polygonal approximation using a competitive Hopfield neural network. Pattern Recognition,27: 1505-1512.
14. Lee, K.C., Funabiki, N.,& Takefuji, Y.(1992). A parallel improvement algorithm for the bipartite subgraph problem. Neural Networks, IEEE Transactions on Volume 3, Issue 1, pp.139 - 145.
15. Chen R.M.,& Huang Y.M.(1998). Multiconstraint task scheduling in multiprocessor system by neural network. In: Proc. IEEE Tenth Int. Conf. on Tools with Artificial Intelligence, Taipei, pp. 288-294.
16. Chen R.M.,& Huang Y.M. (2001). Competitive Neural Network to Solve Scheduling Problem. Neurocomputing, 37(1-4):177-196.
17. Cardeira, C., Mammeri, Z.(1997). Handling Precedence Constraints with Neural Network Based Real-time Scheduling Algorithms. In: Proceedings of Ninth Euromicro Workshop on Real-Time Systems, pp.207 –214.
18. Cardeira, C.,& Mammeri, Z.(1994). Neural networks for multiprocessor real-time scheduling. In: IEEE Proc. Sixth Euromicro Workshop on Real-Time Systems, pp.59-64.
19. Tagliarini, G.A., Christ, J.F.,& Page, E.W.(1991). Optimization Using Neural Networks. IEEE Transaction on Computers , 40(12), 1347-1358.
20. Hopfield, J.J.,& Tank, D.W.(1986). Computing with neural circuits: A model. Science , 233, 625-633.
21. Hopfield, J.J.(1982). Neural networks and physical systems with emergent collective computational abilities. In: Proceedings of the National Academy of Science, 79, pp.2554-2558.

A Distributed Hybrid Algorithm for Optimized Resource Allocation Problem

Kyeongmo Park[1], Sungcheol Kim[2], and Chuleui Hong[2]

[1] School of Computer Science and Information Engineering
The Catholic University, Korea
kpark@catholic.ac.kr
[2] Software School, Sangmyung University,
Seoul, Korea
{sckim, hongch}@smu.ac.kr

Abstract. This paper presents a novel distributed Mean field Genetic algorithm called MGA for the load balancing problems in MPI environments. The proposed MGA is a hybrid algorithm of Mean Field Annealing (MFA) and Simulated annealing-like Genetic Algorithm (SGA). The proposed MGA combines the benefit of rapid convergence property of MFA and the effective genetic operations of SGA. Our experimental results indicate that the composition of heuristic mapping methods improves the performance over the conventional ones in terms of communication cost, load imbalance and maximum execution time. It is also proved that the proposed distributed algorithm maintains the convergence properties of sequential algorithm while it achieves almost linear speedup as the problem size increases.

Keywords: genetic algorithms, mean field annealing, simulated annealing, parallel processing, mapping.

1 Introduction

The load balancing mapping problem is assigning tasks to the processors in distributed memory multiprocessors [1, 2, 3, 4, 5]. Multiple tasks are allocated to the given processors in order to minimize the expected execution time of the parallel program. Thus, the mapping problem can be modeled as an optimization problem in which the interprocessor communication overhead should be minimized and computational load should be uniformly distributed among processors in order to minimize processor idle time. The load balancing is an importance issue in parallel processing.

The proposed Mean Field Genetic Algorithm (MGA) is a hybrid algorithm based on mean field annealing (MFA) [1, 4] and genetic algorithm (GA) [2, 5, 6]. MFA has the characteristics of rapid convergence to the equilibrium state while the simulated annealing (SA) [6, 7] takes long time to reach the equilibrium state. In the proposed method, the typical genetic algorithm is modified where the evolved new states are accepted by the Metropolis criteria as in simulated annealing. The modified Simulate annealing-like Genetic Algorithm is called SGA. The simulation results show that the

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1118 – 1127, 2006.

new MGA is better than MFA and GA, as it reduces inter-processor communication time, load imbalance among processors and expected maximum execution time of the program.

Proposed MGA algorithm takes long time comparing with other mapping algorithm such as MFA and GA, but it must be solved before the execution of a given parallel program in a parallel computer. So the efficient parallel implementation of mapping algorithm is essential for developing parallel programs because the mapping algorithm can be considered as a sequential preprocessing and can be a bottleneck of parallel implementation. We propose two phases of distributed implementation of proposed MGA algorithm. The first phase is for MFA and the second one is for SGA.

2 The Mapping Problem in Multiprocessors

The multiprocessor mapping problem is a typical load balancing optimization problem. A mapping problem can be represented with two undirected graphs, called the Task Interaction Graph (TIG) and the Processor Communication Graph (PCG). TIG is denoted as $G_T(V, E)$. $|V| = N$ vertices are labeled as $(1, 2, \ldots, i, j, \ldots, N)$. Vertices of G_T represent the atomic tasks of the parallel program and its weight, w_i, denotes the computational cost of task i for $1 \le i \le N$. Edge E represents interaction between two tasks. Edge weight, e_{ij}, denotes the communication cost between tasks i and j that are connected by edge $(i, j) \in E$. The PCG is denoted as $G_P(P, D)$. G_P is a complete graph with $|P| = K$ vertices and $|D| = {}_K C_2$ edges. Vertices of the G_P are labeled as $(1, 2, \ldots, p, q, \ldots, K)$, representing the processors of the target multicomputers. Edge weight, d_{pq}, for $1 \le p,q \le K$ and $p \ne q$, denotes the unit communication cost between processor p and q.

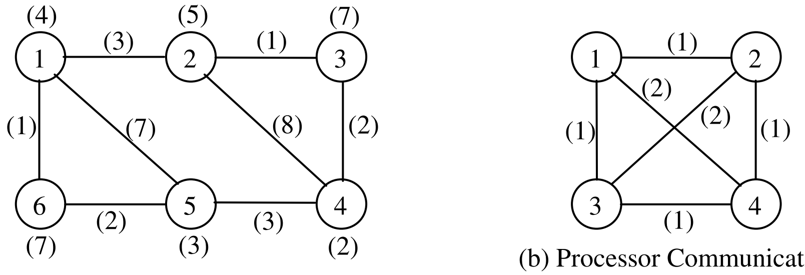
The problem of allocating tasks to a proper processor is to find a many-to-one mapping function M: $V \to P$. That is, each vertex of G_T is assigned to a unique node of G_P. Each processor is balanced in computational load (*Load*) while minimizing the total communication cost (*Comm*) between processors.

$$Comm = \sum_{(i,j)\in E, M(i)\ne M(j)} e_{ij} d_{M(i)M(j)} \cdot \tag{1}$$

$$Load_p = \sum_{i\in V, M(i)=p} w_i , \quad 1 \le p \le K \cdot \tag{2}$$

$M(i)$ denotes the processor to which task i is mapped, i.e. $M(i) = p$ represents that task i is mapped to the processor p. In Equation (1), if tasks i and j in G_T are allocated to the different processors, i.e. $M(i) \ne M(j)$ in G_P, the communication cost occurs. The contribution of this to *Comm* is the multiplication of the interaction amount of task i and j, e_{ij}, and the unit communication cost of different processors p and q, d_{pq}, where $M(i) = p$ and $M(j) = q$. $Load_p$ in Equation (2) denotes the summation of computational cost of tasks i, w_i, which are allocated processor p, $M(i) = p$.

Figure 1 shows an example of the mapping problem. Figure 1(a) represents TIG of N=6 tasks, and Figure 1(b) is for PCG of 2-dimensional mesh topology consisting of K=4 processors. The numbers in circles represent the identifiers of tasks and processor in Figure 1(a) and 1(b) respectively. In Figure 1(a), the weight of vertices and edges is for size of tasks and communications respectively. In Figure 1(b), the weight of edge represents the number of hops between two processors. Figure 2 shows the optimal task allocation to processors on the mapping problem of Figure 1.



(a) Task Interaction Graph (TIG)

(b) Processor Communication Graph (PCG)

Fig. 1. The Example of Mapping Problem

i	1	2	3	4	5	6
$M(i)$	1	2	4	2	1	3

Fig. 2. The Optimal Solution of Figure 1

In MGA, a spin matrix is used to represent the mapping state of tasks to processors. A spin matrix consists of N task rows and K processor columns representing the allocation state. The value of spin element (i, p), s_{ip}, is the probability of mapping task i to processor p. Therefore, the range of s_{ip} is $0 \leq s_{ip} \leq 1$ and the sum of each row is 1. The initial value of s_{ip} is $1/K$ and s_{ip} converges 0 or 1 as solution state is reached eventually. $s_{ip} = 1$ means that task i is mapped to processor p.

Figure 3 displays the initial and final optimal solution spin matrix of Figure 1.

	1	2	3	4
1	0.25	0.25	0.25	0.25
2	0.25	0.25	0.25	0.25
3	0.25	0.25	0.25	0.25
4	0.25	0.25	0.25	0.25
5	0.25	0.25	0.25	0.25
6	0.25	0.25	0.25	0.25

(a) The Initial State

	1	2	3	4
1	1	0	0	0
2	0	1	0	0
3	0	0	0	1
4	0	1	0	0
5	1	0	0	0
6	0	0	1	0

(b) The Solution State

Fig. 3. The Spin Matrix of Figure 1

The cost function, *C*(*s*), is set to minimize the total communication cost of Equation (1) and to equally balance the computational load among processors of Equation (2).

$$C(s) = \sum_{i=1}^{N} \sum_{j \neq i} \sum_{p=1}^{K} \sum_{q \neq p} e_{ij} s_{ip} s_{jq} d_{pq} + r \sum_{i=1}^{N} \sum_{j \neq i} \sum_{p=1}^{K} s_{ip} s_{jp} w_i w_j \, . \tag{3}$$

e_{ij} : The interaction amount of task *i* and *j* in TIG
w_i : The computational cost of task *i* in TIG
d_{pq} : The unit communication cost of processor *p* and *q* in PCG
s_{ip} : The probability of task *i* mapping to processor *p*
r : The ratio of communication to computation cost

The first term of cost function, Equation (3), represents interprocessor communication cost (IPC) between two tasks *i* and *j* when task *i* and *j* are mapped to different processor *p* and *q* respectively. Therefore the first IPC term minimizes as two tasks with large interaction amount are mapped to the same processors. The second term of Equation (3) means the multiplication of computational cost of two tasks *i* and *j* mapped to the same processor *p*. The second computation term also minimizes when the computational costs of each processor are almost the same. It is the sum of squares of the amount of tasks in the same processor. The ratio *r* changes adaptively in the optimization process in order to balance the communication and computation cost. Changing the ratio *r* adaptively results in better optimal solution than fixing the ratio *r*. The optimal solution is to find the minimum of the cost function.

3 Distributed Implementation

3.1 Distributed Mean Field Annealing (MFA)

The mean field annealing (MFA) is derived from simulated annealing (SA) based on mean field approximation method in physics [1]. While SA changes the states randomly, MFA makes the system reach the equilibrium state very fast using the mean value estimated by mean field approximation.

The *N*×*K* spin matrix is partitioned column-wise such that each node is assigned an individual or a group of columns in a spin matrix. A node is a computer system that solves mapping algorithm, while the processor is defined in a target parallel computer. Since in our experiment, the number of nodes, *P*, is generally less than that of processors, *K*, the group of columns in a spin matrix is assigned to each node. However, in real parallel implementation, the number of nodes and that of processors will be same. When task-*i* is selected at random in a particular iteration, each node is responsible for updating its spin value, s_{ip}. The pseudo code for the distributed mean field annealing algorithm of each node is as follows.

```
<Distributed Mean Field Annealing>
while cost change is less than $\varepsilon$ for
      continuous $N$ annealing process begin
   Select a same task-$i$ at random by using same seed
   Compute the local mean field
      $\phi_{ip} = -\sum_{j \neq i}^{N} \sum_{q \neq p}^{K} e_{ij} s_{jq} d_{pq} - r \sum_{j \neq i}^{N} s_{jp} w_i w_j$   for $1 \le p \le K$
   Compute the new spin values at the $i^{th}$ row
      by using global sum operation
      $s_{ip}^{new} = \frac{e^{\phi_{ip}/T}}{\sum_{p=1}^{K} e^{\phi_{ip}/T}}$
   Compute the cost change due to spin updates
      by using global sum operation
      $\Delta C = \sum_{p=1}^{K} \phi_{ip} (s_{ip}^{new} - s_{ip})$
   Update the spin values at the $i^{th}$ row
      $s_{ip} = s_{ip}^{new}$   for $1 \le p \le K$
   Perform global collect
      for a spin value, $s_{ip}$, at the $i^{th}$ row
end
```

In implementing MFA, the cooling schedule has a great effect on the solution quality. Therefore the cooling schedule must be chosen carefully according to the characteristics of problem and cost function. Length of the Markov chain at a certain temperature is the number of state transition to reach the equilibrium state. It is set to the number of state transitions where the cost change is less than =0.5 for continuous *N* annealing process.

3.2 Distributed Simulated Annealing-Like Genetic Algorithm (SGA)

We modified GA such that the new evolved state is accepted with a Metropolis criterion like simulated annealing in order to keep the convergence property of MFA. The modified GA is called SGA. In order to keep the thermal equilibrium of MFA, the new configurations generated by genetic operations are accepted or rejected by the Metropolis Criteria that is used in SA. In the Equation (4), ΔC is the cost change of new state from old state that is made by subtracting the cost of new state from that of old one. *T* is the current temperature.

$$\Pr[\Delta C \text{ is accepted}] = \min\left(1, \exp\left(\frac{\Delta C}{T}\right)\right) \tag{4}$$

A string in the order of tasks whose value is allocated processor identification represents the individual of Genetic Algorithm. For example, a string, "1,3,4,1,2", means that tasks are allocated to processors such that task 1 to processor 1, task 2 to processor 3, task 3 to processor 4, task 4 to processor 1, task 5 to processor 2.

The individuals are generated randomly with the probability as same as that of spin matrix in MFA. For example, if spin values of an arbitrary i^{th} task, which is the elements of i^{th} row, is 0.2, 0.4, 0.1, 0.1, 0.2, an individual is made such that the i^{th} character in a string can be 1 with a probability of 0.2, 2 with that of 0.4, 3 with that of 0.1, 4 with that of 0.1 and so on.

In the experiment, the subpopulation size in each node is set to the number of tasks, *N*. Therefore the size of global population is the multiplication of the number of tasks and the number of nodes, $N \times P$. The linear cost function is chosen as same as that of MFA. The probabilities of crossover and mutation are 0.8 and 0.05 respectively.

In our synchronous distributed genetic algorithm, each node generates subpopulation randomly from the MFA's spin matrix. And then the subpopulation and its fitness value are broadcast to all other nodes and they form the global population. Next, the individuals are selected as much as the size of subpopulation from the global population randomly. Each node executes the sequential genetic algorithm in parallel. Independent genetic operation are implemented and evaluated to its subpopulation. The duration of isolated evolution is called one *epoch* and the *epoch length* is the number of predefined generations for a node before synchronizing communication among the nodes. The epoch length is set to the *N/P*, where *N* is the number of tasks and *P* is the number of nodes. *max_epoch* is the number of synchronous communications. It is set to *P*.

The pseudo code for the distributed genetic algorithm of each node is as follows.

```
<Distributed SGA>
Initialize subpopulation($P_{sub}$) from MFA spin matrix
for iteration is less than max_epoch begin
   Calculate fitness for $P_{sub}$
   for generations = 1 until epoch_length begin
      Select individuals from subpopulation
      Reproduce next population
      for select 2 individuals by turns begin
         Perform crossover with probability of crossover
         Calculate the cost change ($\Delta C$)
         if $\exp(-\Delta C/T) >$ random[0,1] then
            Accept new individuals
      end
      for all individuals begin
         Perform mutation with probability of mutation
         Calculate the cost change ($\Delta C$)
         if $\exp(-\Delta C/T) >$ random[0,1] then
            Accept new individuals
      end
   end
   broadcast $P_{sub}$ to all other nodes;
   select new $P_{sub}$ randomly;
   Keep the best individual
end
```

3.3 MGA Hybrid Algorithm

A new hybrid algorithm called MGA combines the merits of mean field annealing (MFA) and simulated annealing-like genetic algorithm (SGA). MFA can reach the thermal equilibrium faster than simulated annealing and GA has powerful and various genetic operations such as selection, crossover and mutation.

First, MFA is applied on a spin matrix to reach the thermal equilibrium fast. After the thermal equilibrium is reached, the population for GA is made according to the distribution of task allocation in the spin matrix. Next, GA operations are applied on the population while keeping the thermal equilibrium by transiting the new state with Metropolis criteria. MFA and GA are applied by turns until the system freeze. The followings are the pseudo code for the distributed MGA algorithm of each node.

```
<Distributed MGA Hybrid Algorithm>
Initialize mapping problems /* getting TIG and PCG */
Forms the spin matrix, $s=[s_{11}, …, s_{ip}, …, s_{NK}]$
Set the initial ratio $r$
Get the initial temperature $T_0$ , and set $T= T_0$
while $T \geq T_f$ begin
   Executes MFA
   Forms GA population from a spin matrix of MFA
   Executes SGA
   Forms the spin matrix of MFA from GA population
   Adjusts the ratio $r$
   $T= \alpha \times T$   /*decrease the temperature*/
end
```

Initial temperature, T_0, is set such that the probability where the cost change is less than (=0.5) is more than 95% for the number of tasks (N) annealing process. Final temperature (T_f) is set to the temperature where the value of the cost change is in /1,000 for continuous N temperature changes. A fixed decrement ratio, α, is set to 0.9 experimentally. This strategy decreases the temperature proportional to the logarithm of the temperature.

4 Simulation Results

The proposed MGA hybrid algorithm is compared with MFA and GA. In this simulation, the size of tasks is 200 and 400 (only the results of task size of 400 are shown). The multiprocessors are connected with wrap-around mesh topology. The computational costs of each task are distributed uniformly ranging [1..10]. The communication costs between any two tasks ranges [1..5] with uniform distribution. The number of communications is set to 1, 2, or 3 times of the number of tasks. The experiment is performed 20 times varying the seed of random number generator and TIG representing the computational and communication cost.

The coefficient r in the linear cost function is for balancing the computation and communication cost between processors. The initial ratio r is computed as in Equation (4). As the temperature decreases, the coefficient r varies adaptively according to

Equation (5) in order to reflect the changed interprocessor communication cost. r_{old} is the ratio used at the previous temperature and r_{new} is the newly calculated ratio at the current temperature.

$$r_{init} = \frac{\text{Comm. cost}}{\text{\# of processors} \times \text{Comp. cost}} = \frac{\sum_{i=1}^{N} \sum_{j \neq i} \sum_{p=1}^{K} \sum_{q \neq p} e_{ij} s_{ip} s_{jq} d_{pq}}{K \times \sum_{i=1}^{N} \sum_{j \neq i} \sum_{p=1}^{K} s_{ip} s_{jp} w_i w_j} \tag{4}$$

$$r_{new} = \begin{cases} 0.9 \times r_{old} & \text{if } r_{new} < r_{old} \\ r_{old} & \text{Otherwise} \end{cases} \tag{5}$$

Table 1. The maximum completion times when the initial value of *r* is fixed or *r* varies adaptively according to Equation (5)

Problem Size (N = 400)		MFA		MGA	
|E|	K	fixed *r*	variable *r*	fixed *r*	variable *r*
400	16	627.1	279.05	256.4	**222.75**
800	16	1189.95	888.1	617.15	**587**
1200	16	1862.8	1557.4	**971.9**	987.5
400	36	410.5	152.85	143.85	**128.65**
800	36	834.45	617	410.9	**385.15**
1200	36	1376.8	1065.5	714.65	**692.95**

Table 1 compares the maximum completion times when the initial value of *r* is fixed or *r* varies adaptively according to Equation (5). The completion time of an arbitrary processor is the sum of computation costs of its processor and the communication cost with other processors. The maximum completion time is defined as the maximum value among the completion times of all processors. In Table 1, *N* is the number of tasks, |*E*| is the total number of interprocessor communications, and *K* represents the number of processors.

Table 2. Total interprocessor communication cost, Percent computational cost imbalance, and Execution Time for Problem size 400

		Total Comm. Time			Comp. Cost. Imbalance			Exec. Time (secs.)		
|E|	K	MFA	GA	MGA	MFA	GA	MGA	MFA	GA	MGA
400	16	994.3	2370.5	**629.7**	47%	37%	**32%**	**26.2**	61.7	95.6
800	16	8089.1	6714.6	**4004.2**	61%	**37%**	57%	**40.7**	70.3	150.9
1200	16	14677	11743	**8348.4**	49%	**35%**	54%	**51.7**	63.7	190.9
400	36	1062.7	3539.5	**852.4**	60%	59%	**50%**	94.9	**77.6**	212.4
800	36	10021	10360	**5603**	79%	**62%**	72%	122.9	**76.4**	260.6
1200	36	19937	17780	**11868**	75%	**60%**	70%	148.1	**70.6**	317.5

Table 2 displays average total interprocessor communication cost of each algorithm. The average performance improvement form MFA to MGA, which is the percent reduction of communication cost normalized by that of MFA, is 33%. It is 45% from GA to MGA. It also displays computational cost imbalance, which is defined as the difference between maximum and minimum computational cost of processors normalized by the maximum cost.

The computational cost imbalance of each algorithm displays a little difference, while total communication costs in Table 2 are much more different. This implies that the interprocessor communication cost has a greater effect on the solution quality than the computational cost. Finally the last column of Table 2 displays the execution time of each algorithm. The averaged execution time of MGA is 1.5 and 1.7 times longer than that of MFA and GA respectively. This is a trade-off between the solution quality and execution time.

The proposed MGA takes a long time compared with other heuristic algorithm. So we proposed the efficient distributed implementation of MGA. Fortunately, the both of MFA and GA can be implemented in parallel inherently. The parallel speedup generally increases proportional to the problem size due to reducing synchronization cost (Figure 4). We can find that the proposed distributed algorithm maintains the solution quality of sequential algorithm.

The simulation is implemented in MPI environments that are made up of 600Mhz personal computers running Linux operating system connected via 10Mbps Ethernet.

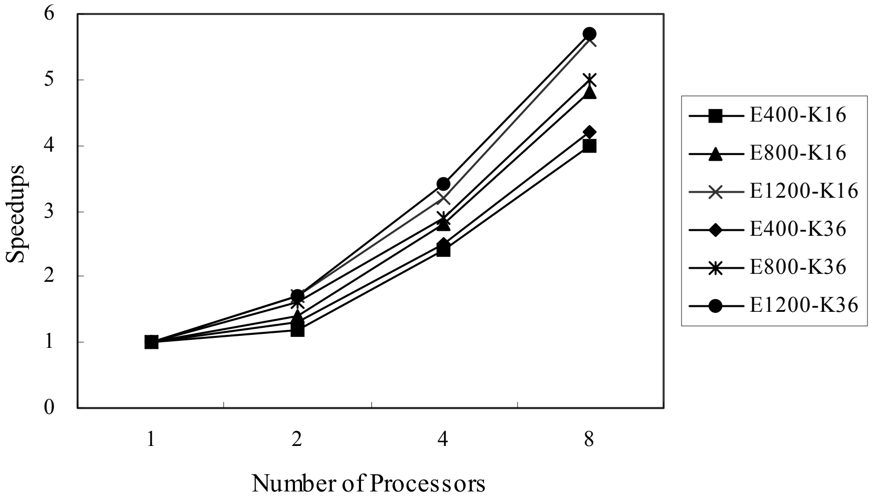


Fig. 4. Speedups of Different Problems

5 Conclusions

In this paper, we proposed a new hybrid algorithm called MGA. The proposed approach combines the merits of MFA and GA on a load balance problem in distributed memory multiprocessor systems. The solution quality of MGA is superior to that of

MFA and GA while execution time of MGA takes longer than the compared methods. There can be the trade off between the solution quality and execution time by modifying the cooling schedule and genetic operations. MGA was also verified by producing more promising and useful results as the problem size and complexity increases. The proposed algorithm can be easily developed as a distributed algorithm since MFA and GA can be parallelized easily. This algorithm also can be applied efficiently to broad ranges of NP-Complete problems.

References

1. Bultan, T., Aykanat, C. : A New Mapping Heuristic Based on Mean Field Annealing. Journal of Parallel & Distributed Computing,16 (1992) 292-305
2. Heiss, H.-U., Dormanns, M. : Mapping Tasks to Processors with the Aid of Kohonen Network. Proc. High Performance Computing Conference, Singapore (1994) 133-143
3. Park, K., Hong, C.E. : Performance of Heuristic Task Allocation Algorithms. Journal of Natural Science, CUK, Vol. 18 (1998) 145-155
4. Salleh, S., Zomaya, A. Y.: Multiprocessor Scheduling Using Mean-Field Annealing. Proc. of the First Workshop on Biologically Inspired Solutions to Parallel Processing Problems (BioSP3) (1998) 288-296
5. Zomaya, A.Y., Teh, Y.W.: Observations on Using Genetic Algorithms for Dynamic Load-Balancing. IEEE Transactions on Parallel and Distributed Systems, Vol. 12, No. 9 (2001) 899–911
6. Hong, C.E.: Channel Routing using Asynchronous Distributed Genetic Algorithm. Journal of Computer Software & Media Tech., SMU, Vol. 2. (2003)
7. Hong, C., McMillin, B.: Relaxing synchronization in distributed simulated annealing. IEEE Trans. on Parallel and Distributed Systems, Vol. 16, No. 2. (1995) 189-195

A Swarm Optimization Model for Energy Minimization Problem of Early Vision

Wenhui Zhou[1], Lili Lin[2], and Weikang Gu[1]

[1] Department of Information Science and Electronic Engineering, ZheJiang University, HangZhou, 310027, China
sunshine@hzcnc.com
[2] College of Information and Electronic Engineering, ZheJiang Gongshang University, Hangzhou, 310035, China

Abstract. This paper proposes a swarm optimization model for energy minimization problem of early vision, which is based on a multi-colony ant scheme. Swarm optimization is a new artificial intelligence field, which has been proved suitable to solve various combinatorial optimization problems. Compared with general optimization problems, energy minimization of early vision has its unique characteristics, such as higher dimensions, more complicate structure of solution space, and dynamic constrain conditions. In this paper, the vision energy functions are optimized by repeatedly minimizing a certain number of sub-problems according to divide-and-conquer principle, and each colony is allocated to optimize one sub-problem independently. Then an appropriate information exchange strategy between neighboring colonies, and an adaptive method for dynamic problem are applied to implement global optimization. As a typical example, stereo correspondence will be solved using the proposed swarm optimization model. Experiments show this method can achieve good results.

1 Introduction

It is well know that many early vision problems are generally required to assign a label from some finite sets $\mathcal{L}$ to each pixel. For motion or stereo, the labels are disparities. For image restoration they represent intensities. While for edge detection or image segmentation, the labels are binary variables. These labeling problems can always be formulated in terms of energy minimization in MAP-MRF framework. One advantage of using energy functions is many constraints can be added naturally. The standard energy function can be formulated into a data term E_{data} and smoothness term E_{smooth}, as shown in Eq.(1) [1], [2]

$$E\left(f\right) = \sum_{p} E_{data}\left(f_p\right) + \lambda \sum_{p,q} E_{smooth}\left(f_p, f_q\right) \tag{1}$$

where $f = \{f_p | p \in \mathcal{P}\}$ is the labeling of image $\mathcal{P}$. $E_{data}(f_p)$ is a data penalty term for pixel p assigned with label f_p, and $E_{smooth}(f_p,\ f_q)$ is a smoothness term that imposes punishment if neighboring pixels have been assigned different

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1128–1137, 2006.

labels. Apparently, it is an NP-hard problem to minimize the energy function in Eq.(1).

In recent years, swarm optimization has become a new artificial intelligence field inspired by the behavior of insect swarms. These insects exhibit surprising social behaviors with very simple individuals. Since the behaviors of individuals are independent, swarm optimization is a good candidate for parallelization. As a typical technique of the swarm optimization, the ant colony optimization (ACO) has been proved suitable to solve various complex optimization problems [3], [4]. In particular, ACO algorithms have been shown to be very efficient to solve the traveling salesman problem (TSP). However, TSP is static optimization problem, i.e. the constraint conditions of the problem do not change with time.

Compared with general optimization problems, energy minimization of early vision has its unique characters, such as higher dimensions, more complicate structure of solution space, and dynamic constrain conditions. This paper proposes a swarm optimization model for energy minimization problem of early vision, which is based on multi-colony ant scheme. To solve a complex optimization problem, we generally follow the divide-and-conquer principle. Therefore the energy function in Eq.(1) can be broke up into a number of sub-problems, each colony is allocated to optimize one sub-problem independently. Then an appropriate information exchange strategy between neighboring colonies, and an adaptive method for dynamic problem are applied to implement global optimization. As a typical example, stereo correspondence will be solved using the proposed swarm optimization model. Experiments show this method can achieve good results.

2 Ant Colony Optimization

ACO was inspired by the foraging behavior of real ant colony. Dorigo et al. [3], [4] discovered the key factor of ant foraging behavior is a chemical substance called pheromone deposited by the ants that found the source of the food, which can guide other ants to the food source. Moreover, the pheromone evaporates gradually with time, which means the shorter the path is, the less quantity of the pheromone evaporates, and the higher probability of this path will be chosen by subsequent ants. So the pheromone on this path is enhanced. This indirect communication between the ants via the pheromone trails and positive feedback allow ants to find the shortest path between their nest and food source quickly.

2.1 Basic Ant Colony System

Dorigo et al. successfully exploited the foraging behavior of real ant colonies in artificial ant colonies to solve discrete optimization problems. In general, the basic ant colony system algorithm can be implemented by iterating the following two steps:

1) Each ant chooses next solution component according to transition probabilities.

The transition probability from the current solution component i to the next solution component j of ant k is defined as follows.

$$p_{ij}^{k}\left(t\right)=\begin{cases}\frac{\tau_{ij}^{\alpha}(t)\cdot\eta_{ij}^{\beta}}{\sum_{l\in N_{i}^{k}}\tau_{il}^{\alpha}(t)\cdot\eta_{il}^{\beta}}, & if\ j\in N_{i}^{k}\\ 0, & otherwise\end{cases}\tag{2}$$

where η_{ij} is the heuristic information which usually is inversely proportional to the distance, τ_{ij} is the pheromone trail between the solution component i and j, N_i^k is a set of feasible solution components, α and β determine the relation between pheromone and heuristic information.

2) Pheromone trail update

Pheromone trail update includes two elements: evaporation and deposition. In general, the following pheromone update rule is used,

$$\tau_{ij}\left(t\right)\leftarrow\left(1-\rho\right)\cdot\tau_{ij}\left(t\right)+\rho\cdot\Delta\tau_{ij}\left(t\right)\tag{3}$$

where ρ is the local pheromone decay parameter, $0<\rho<1$. $\Delta\tau_{ij}(t)$ is the new amount of pheromone deposited by ants in the current iteration.

2.2 Max-Min Ant System

Max-Min Ant System (MMAS) developed by Stutzle and Hoos in 1996 [5], [6] is one of the best performing ACO algorithms for combinatorial optimization problems. The MMAS is different from Ant System in three main aspects,

1. To exploit the best solution found during an iteration or during the run of the algorithm, after each iteration only the iteration-best or the global-best ant is used for pheromone update.
2. In order to avoid the stagnation of the search, the pheromone value of every edge is limited in the interval of $[\tau_{min}, \tau_{max}]$.
3. The pheromone trails are initialized to τ_{max}, which can lead to a higher exploration of tours at the start of the algorithm.

3 Swarm Optimization Model for Vision Problems

3.1 Energy Minimization of Early Vision Problems

It is usually quite difficult to minimize the energy functions of early vision, because their solution space is often thousands of dimensions and have thousands of local minima. According to the divide-and-conquer principle, we need break up the energy minimization problem into a number of sub-problems that can be optimized independently. Since the complexity of vision problems and the variety of constraint conditions, it is impossible to express energy minimization of early vision problem as the sum of the solutions of many independent sub-problems without containing any same variables between each other.

To be specific, let $E_i(f_i)$ is the ith sub-function with the set of variables f_i, and $E_{ij}(f_i, f_j)$ is the interactive term between the ith and jth sub-functions, then the energy function in Eq.(1) can be formulated as follows.

$$\min_f \{E(f)\} = \sum_i \min_{f_i} \{E_i(f_i)\} + \sum_i \sum_{j \neq i} \min_{f_i, f_j} \{E_{i,j}(f_i, f_j)\} \tag{4}$$

Since it is very difficult to divide Eq.(4) into independent sub-problems directly, we use the following continuous approximate formula by introducing state variables $s_i(t)$.

$$\min_f \{E(f,t)\} \approx \sum_i \min_{f_i} \{E'_i(f_i, t)\} \tag{5}$$

where $E'_i(f_i, t) = \min_{f_i} \{E_i(f_i) + E_i(f_i, s_j(t - \Delta t))\}$ is the ith independent sub-problem at the instant$t$, and $s_j(t - \Delta t) = \min_{f_j} \left\{E'_j(f_j, t - \Delta t)\right\}$ is the states of the jth independent sub-problem at the instant $t - \Delta t$, which are the optimization results of f_j at the instant $t - \Delta t$, and they can be regarded as constants at the instant t. $E_i(f_i, s_j(t - \Delta t))$ can be regarded as the constraints of the ith independent sub-problem at the instant$t$. Apparently, when $\Delta t \rightarrow 0$ and $t \rightarrow \infty$, Eq.(5) is equivalent to Eq.(4).

The discrete form of the Eq.(5) can be formulated as follows.

$$\min_f \{E(f,k)\} \approx \sum_i \min_{f_i} \{E'_i(f_i, k)\} \tag{6}$$

where $E'_i(f_i, k) = \min_{f_i} \{E_i(f_i) + E_i(f_i, s_j(k-1))\}$,
$s_j(k-1) = \min_{f_j} \left\{E'_j(f_j, k-1)\right\}$, and k is the number of iteration.

Therefore, we can minimize the energy function in Eq.(1) by an iterative process of minimizing the sub-problems $E'_i(f_i, k)$. Moreover, because of the existence of interactive terms, the constraints of each sub-problem are different during each iteration, that is each sub-problem minimization is not a static optimization problem.

3.2 Dynamic Multi-colony Ant Model

$E'_i(f_i, k)$ is an independent sub-problem during the kth iteration, which can be optimized by an ant colony system. While the state update in each sub-problem should be implemented by information exchange between colonies. One of this paper's main motives is to present an appropriate pheromone exchange strategies between the colonies. Middendorf et al. [7] considered the colonies should not exchange too much information and too often, and proposed four strategies for information exchange. However, their researches are based on the fact all colonies find good solutions for the same optimization problem.

In our problems, each colony is allocated to optimize one sub-problem, and the constraints of each sub-problem are different, i.e., the problems optimized by each colony are different. Therefore, inspired by the results of Middendorf et al., we propose the following information exchange strategy.

1. Information exchange only takes place between the neighboring colonies, and an information exchange is done every $k$generations. Each colony should hold relative stable solution before information exchange between each other.
2. During each information exchange step, every colony sends its local best solution to its neighboring colonies.
3. Since the constraints of problem optimized by each colony are different, the exchanged information of neighboring colonies is only suggestive. Therefore, the exchanged information is weighted by a fading parameter γ, $0<\gamma<1$. According to this viewpoint, "information diffusion" is a more suitable expression than "information exchange" in this instance.

Furthermore, since the constraints (states value) of sub-problem will be changed with the information diffusion between neighboring colonies, the problems optimized by each colony are dynamic rather than static. Fortunately, we can know when the constraints are changed. D. Angus et al. [8] proposed an adaptive method based on pheromone normalization. In their method, the pheromone value $\tau_{i,j}$ between city i and j is replaced by $\tau_{i,j}/\tau_{imax}$, where τ_{imax} is the maximum pheromone value on any edge of city i. In this paper, we use their method to normalize pheromone values after information diffusion. But considering the range of pheromone in MMAS, $\tau_{max}$is multiplied after pheromone normalization in our algorithm.

4 Experiments

As a typical instance of the early vision problems, stereo correspondence is used to prove the performance of the proposed swarm optimization model.

4.1 Stereo Correspondence

Stereo correspondence is one of the key problems of stereo vision, which finds a unique mapping between the pixels belonging to stereopsis of the same scene. If the images are rectified, then the corresponding pixels should lie on the same horizontal scan-line. The difference in the horizontal position of corresponding pixels is termed as disparity. In this paper, we assume the stereopsis is already rectified.

In stereo correspondence, label set f in Eq.(1) corresponds to the range of disparity d. Let l_i is the pixel set on the ith scan-line, then Eq.(1) can be rewritten as follows,

$$\begin{aligned} E(d) = \sum_{i} \left\{ \sum_{p \in l_i} E_{data}(d_p) + \lambda_1 \sum_{p,q \in l_i} E_{smooth}^{intra}(d_p, d_q) \right\} \\ + \lambda_2 \sum_{i} \sum_{j,\ i \neq j} \left\{ \sum_{m \in l_i, n \in l_j} E_{smooth}^{inter}(d_m, d_n) \right\} \end{aligned} \tag{7}$$

where E_{smooth}^{intra} is the smoothness term between neighboring pixels on the same scan-line, and E_{smooth}^{inter} is the smoothness term between two neighboring scan-lines.

According to Eq.(5) and (6), Eq.(7) can be reformulated as the following iterative discrete form,

$$E\left(d,k\right) \approx \sum_{i} E_{l_i}\left(d_{p\in l_i},k\right) \tag{8}$$

$$\begin{aligned} E_{l_i}\left(d_{p\in l_i},k\right) &= \sum_{p\in l_i} E_{data}\left(d_p\right) + \lambda_1 \sum_{p,q\in l_i} E_{smooth}^{intra}\left(d_p,d_q\right) \\ &+ \lambda_2 \sum_{j,\ i\neq j} \left\{ \sum_{m\in l_i,n\in l_j} E_{smooth}^{inter}\left(d_m,s_n\left(k-1\right)\right) \right\} \end{aligned} \tag{9}$$

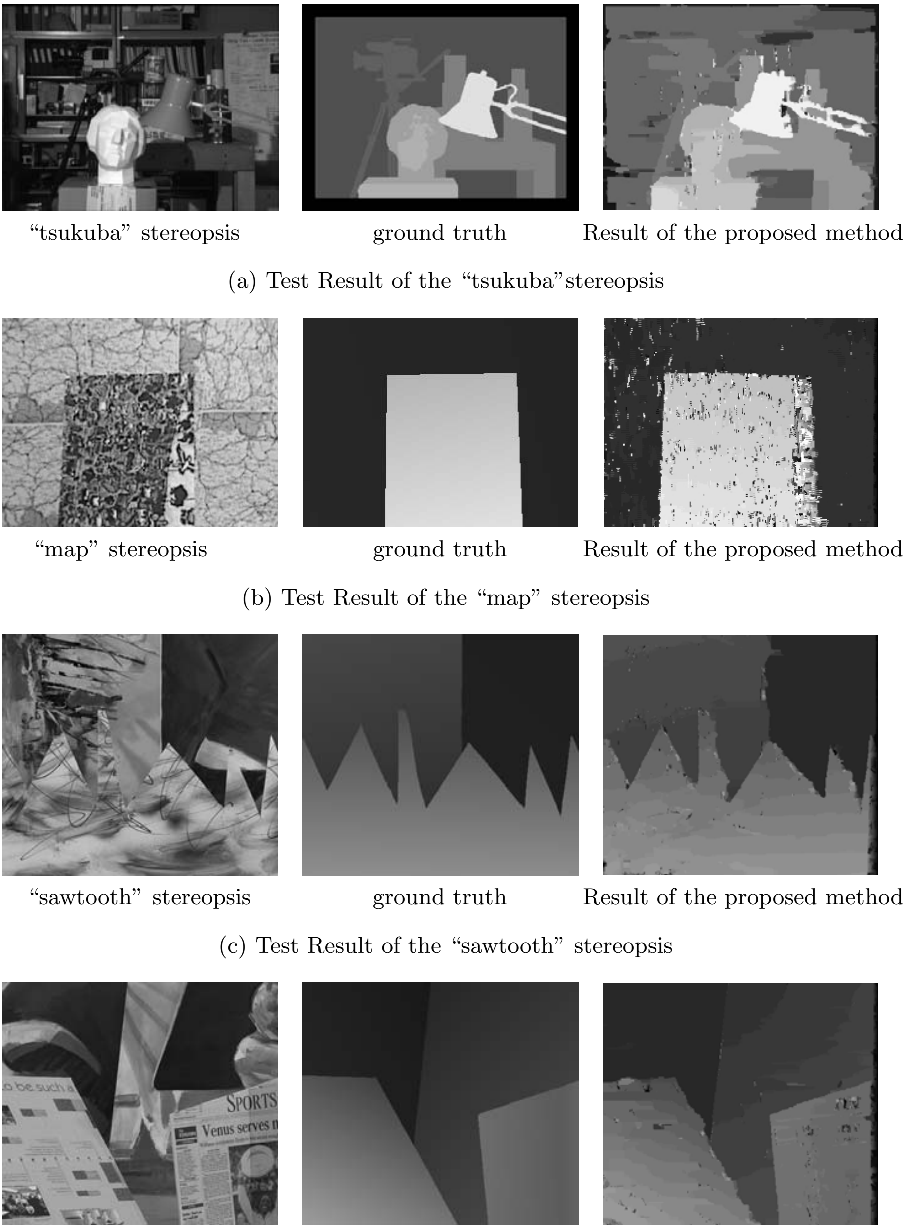
where $s_n(k\text{-}1)$ is the state of pixel n which is the disparity result of pixel n during the (k-1)th iteration, and it can be regarded as a constant during the kth iteration.

In dynamic multi-colony ant model, each colony is allocated to minimize one sub-problem, which is a 1D optimization problem based on the disparity space image (DSI). Let directed weighted graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ represents DSI of the ith scan-line. All possible disparity values of all pixels on the ith scan-line, the source s, and the sink t form the vertices set $\mathcal{V}$. Only vertices of the neighboring pixels are connected by directed edges, and vertices of the same pixels have no connections between each other. The "directed" is defined as the direction from left to right on the scan-line of the right image. The costs of edges are derived from the energy function in Eq.(8). Data term uses the absolute difference of the luminance of corresponding pixels, and the Potts model is chosen as the smooth term.

Initially, the pheromone on each edge is set to their maximally possible value τ_{max}, and the heuristic information of each edge is the reciprocal of the energy value. Each ant starts from the source s, and chooses the next vertex according to Eq.(2). In order to restrict the feasible solution components of subsequent pixels within a narrower range, the ant's tabu table is updated according to uniqueness constraint when it arrives at one vertex. After each iteration, only the local best ant is used for pheromone update. Then information exchange between the colonies is implemented according to the strategies in Section 3.2.

4.2 Results Analysis and Conclusions

Test Results of Standard Test Stereo. Four pairs of standard test stereo images obtained from the Middlebury College's stereo vision research website [9] are chosen for the performance experiments. The parameter values in energy function used in experiments are λ_1=λ_2=5, and the parameter values in ant colony system are α=1, β=5, ρ=0.9, γ=1, k=10, the range of pheromone in



“tsukuba” stereopsis ground truth Result of the proposed method

(a) Test Result of the “tsukuba”stereopsis

“map” stereopsis ground truth Result of the proposed method

(b) Test Result of the “map” stereopsis

“sawtooth” stereopsis ground truth Result of the proposed method

(c) Test Result of the “sawtooth” stereopsis

“venus” stereopsis ground truth Result of the proposed method

(d) Test Result of the “venus” stereopsis

Fig. 1. Results of stereo correspondence using the proposed swarm optimization model

Table 1. Solution space parameters of each sub-problem of the stereo pairs in Fig.1

stereopsis	*PixelNum*	*DPMax*	Edge number
"tsukuba"	388	16	99,088
"map"	284	32	289,824
"sawtooth"	434	32	443,424
"venu"	434	32	443,424

MMAS is from 0.01 to 0.9. The number of ants and iterations are 500 and 300, respectively.

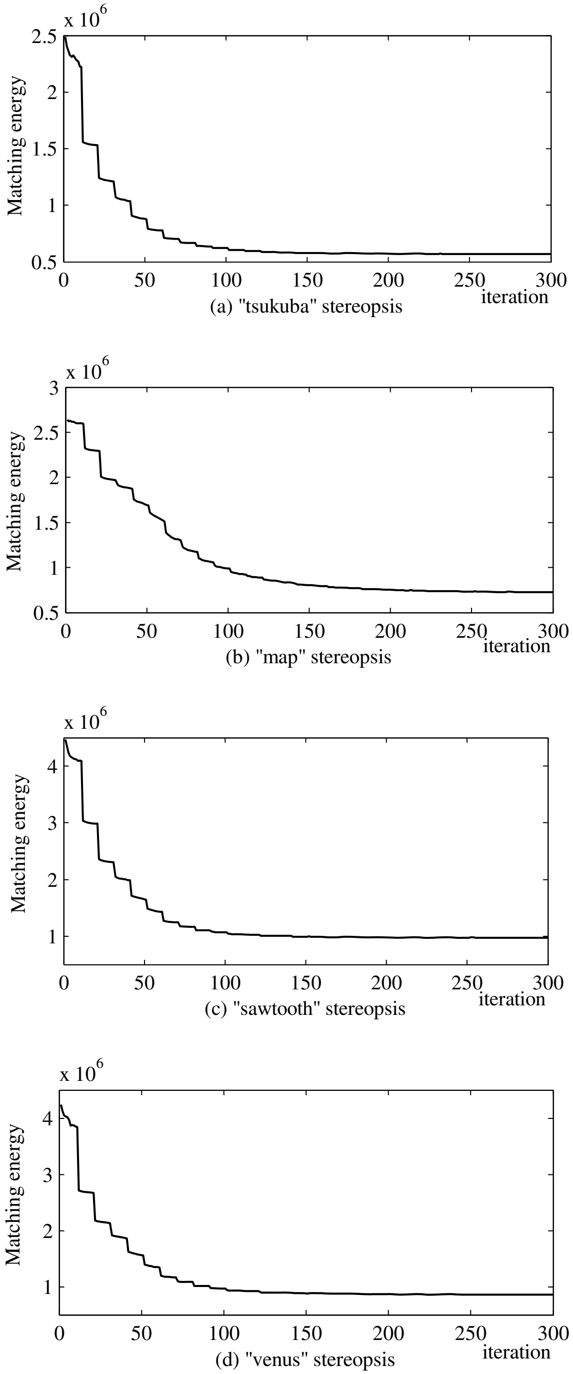
The images in the left, middle and right columns of Fig.1 are the right images of the stereo pairs, the ground truth, and results of the proposed method, respectively.

According to Section 4.1, the edge number of each sub-problem is about $(\mathit{PixelNum}\text{-}1)\times \mathit{DPMax}^2+\mathit{DPMax}$, where *PixelNum* is the number of pixels on a scanline, and *DPMax* is the range of disparity search. And the size of the search space is about $\mathit{DPMax}^{\mathit{PixelNum}}$. The solution space parameters of each sub-problem of the stereo pairs in Fig.1 are given in Table 1.

Compared with reference [10] where 3000 ants are employed to deal with 100×100 size images, and reference [11] where 1500 ants are used to extract edge of 512×512 size images, it is obvious only 500 ants is not enough for the stereo correspondence in this paper. Considering this fact, the results in Fig.1 is acceptable although they are not as good as those of the best method on the Middlebury College's stereo vision research website [9].

Analysis Curves of Matching Energy. According to Eq.(8), the matching energy of the whole image is sum of the minimum solution of each sub-problem. Fig.2 shows the curve of matching energy versus the number of iterations. These figures indicate when information diffusion occurs, there is a small fluctuation on the curve, which can make the matching energy escape from the local minimum. And during the other iterations, the matching energy is monotonically decreasing. So the curve is decreasing as a whole. After 200 iterations, the curve decreases slowly and approaches flat that means the proposed swarm optimization model has better convergence performance.

Conclusions. This paper proposes a swarm optimization model based on a dynamic multi-colony ant scheme for energy minimization problem of early vision. According to divide-and-conquer principle, each colony is allocated to optimize one sub-problem of vision energy function independently. Then we present an appropriate information exchange strategy and an adaptive method for dynamic problem. As a typical example, experiments of stereo correspondence show this method can achieve good results.



(a) "tsukuba" stereopsis

(b) "map" stereopsis

(c) "sawtooth" stereopsis

(d) "venus" stereopsis

Fig. 2. Matching energy curves of the standard test stereopsis

References

1. Boykov, Y., Veksler, O., and Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence. **23(11)** (2001) 1222–1239
2. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? In European Conference on ComputerVision. **3** (2002) 65–81
3. Dorigo, M., Maniezzo, V.: Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics - Part B. **26(1)** (1996) 29–41
4. Dorigo, M., Gambardella, L.M.: Ant Colony System: A cooperative learning approach to the travelling salesman problem. IEEE Transactions on Evolutionary Computation. **1(1)** (1997) 53–66
5. Stutzle, T., Hoos, H.H.: The MAX-MIN Ant System and Local Search for the Traveling Salesman Problem, Proceedings of ICEC'97. (1997) 309–314
6. Stutzle, T., Hoos, H.H.: Improvements on the Ant System: Introducing the MAX-MIN Ant System. Artificial Neural Networks and Genetic Algorithms. (1998) 245–249
7. Middendorf, M., Reischle, F., and Schmeck, H.: Multi Colony Ant Algorithms, Journal of Heuristics. **8** (2002) 305–320
8. Angus, D., Hendtlass, T.: Dynamic Ant Colony Optimisation, Applied Intelligence. **23** (2005) 23–38
9. http://www.middlebury.edu/stereo.
10. Ramos, V., Almeida, F.: Artificial Ant Colonies in Digital Image Habitats - A Mass Behaviour Effect Study on Pattern Recognition, Proceedings of ANTS2000 - 2nd International Workshop on Ant Algorithms, Brussels, Belgium. (2000) 113–116
11. Nezamabadi-pour, H., Saryazdi, S., and Rashedi, E.: Edge Detection using Ant Algorithms", in Soft Computing - A Fusion of Foundations, Methodologies and Applications, Springer-Verlag GmbH, (2005)

PSO-Based Hyper-Parameters Selection for LS-SVM Classifiers

X.C. Guo[1,2], Y.C. Liang[1,*], C.G. Wu[1,3], and C.Y. Wang[1]

[1] College of Computer Science and Technology, Jilin University, Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education,
Changchun 130012, P.R. China
ycliang@jlu.edu.cn

[2] College of Science, Northeast Dianli University, Jilin 132012, China

[3] The Key Laboratory of Information Science & Engineering of Railway ministry/The Key Laboratory of Advanced information science and network technology of Beijing,
Beijing Jiaotong University, Beijing 100044, China

Abstract. The determination for hyper-parameters including kernel parameters and the regularization is important to the performance of least squares support vector machines (LS-SVMs). In this paper, the problem of model selection for LS-SVMs is discussed. The particle swarm optimization (PSO) is introduced to select the LS-SVMs hyper-parameters. In the proposed method we do not need to consider the analytic property of the generalization performance measure and the number of hyper-parameters. The feasibility of this method is evaluated on benchmark data sets. Experimental results show that better performance can be obtained. Moreover, different kinds of kernel families are investigated by using the proposed method. Experimental results also show that the best and good test performance could be obtained by using the SRBF and RBF kernel functions, respectively.

Keywords: least squares support vector machines; particle swarm optimization; fitness function; parameter selection; classification.

1 Introduction

Support vector machines (SVMs) were developed by Vapnik and his colleagues [1]. SVMs are based on the structural risk minimization principle (SRM), which has been shown to be superior to the traditional empirical risk minimization principle (ERM) employed by conventional neural networks. SRM minimizes an upper bound of generalization error as opposed to ERM that minimizes the error on training data. Therefore, the solution of SVM may be global optimum while other neural network models tend to fall into a local optimal solution, and overfitting is unlikely to occur with SVM [2, 3, 4]. The classical training algorithm of SVMs is equivalent to solving a quadratic programming with linearly constraints. During the last decade, many pattern recognitions have been tackled using SVMs. Least Squares Support Vector

[*] Corresponding author.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1138 – 1147, 2006.

Machines (LS-SVMs) are introduced by Suykens et. al as reformulations to standard SVMs [5] which lead to solving linear Karush-Kuhn-Tucker (KKT) systems for classification problems as well as regression. LS-SVM simplifies the solution process of standard SVM in a great extent by substituting the inequality constraints by equality counterparts. Consequently, the decision function can be obtained by solving a group of linear equalities rather than quadratic programming.

For the standard SVMs and its reformulations, LS-SVM, the regularization parameter and kernel parameter(s) are called hyper-parameters, which play a crucial role to the performance of the SVMs. There exist different techniques for tuning the hyper-parameters related to the regularization constant and the parameter of kernel function. These methods can be divided into two classes: one is the analytical and algebraic techniques, another is heuristic search algorithm (including grid search). The analytical and algebraic techniques are almost based on the gradient of some generalized error measure [6-13]. Recently, genetic algorithm, simulated annealing algorithm and other evolutionary strategy [14-19] are employed for the hyper-parameters of SVMs. Iterative gradient-based algorithms, which usually rely on smoothed approximations of a function, do not ensure that the search direction points exactly to an optimum of the generalization performance measure which is often discontinuous. Grid search which needs an exhaustive search over the space of hyper-parameters is often used to select parameters [20]. This procedure requires a grid search over the space of parameter values and needs to locate the interval of feasible solution and a suitable sampling step. This is a tricky task since a suitable sampling step varies from kernel to kernel and the grid interval may not be easy to locate without prior knowledge of the problem. Moreover, when there are more than two hyper-parameters, the manual model selection may become intractable.

In this paper, a new parameters selection algorithm is proposed based on the principles of the particle swarm optimization (PSO). The PSO is an evolutionary computation technique based on swarm intelligence. It follows a collaborative population-based search, which models over the social behavior of bird flocking. The PSO system combines experiences form both self and neighboring and attempts to balance exploration and exploitation. The PSO has many advantages over other heuristic techniques, e.g., it can be used effectively to exploit the distributed and parallel computing capabilities, to escape local optima, and to implement in a few lines of computer codes. The proposed method is applied to tuning kernels and regularization parameters of LS-SVMs.

2 LS-SVM Classifiers

Consider a given training set $\{(x_i, y_i) \mid x_i \in R^n, y_i \in \{-1,+1\}\}_{i=1}^{N}$, where x_i is input and y_i is the binary class label. The discriminant function takes the following form:

$$y = \text{sign}[w^T \phi(x) + b] \tag{1}$$

where the nonlinear function $\phi(\cdot)$, which is not explicitly constructed, maps the input into a higher dimensional feature space (can be infinite dimension). The coefficient vector w and bias term b need to be determined. In order to obtain the coefficient vector w and bias term b, the following optimization problem to be solved is as follows [5, 20]

$$\min_{w,e_i} J(w,e) = \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^{N} e_i^2 \tag{2}$$

subject to the equality constraints

$$y_i[w^T \phi(x_i) + b] = 1 - e_i, \; i = 1,2,\cdots,N \tag{3}$$

The Lagrangian corresponding to Eq. (2) can be defined as:

$$L(w,b,e_i,\alpha) = J(w,b,e) - \sum_{i=1}^{N} \alpha_i \{ y_i [w^T \phi(x_i) + b] - 1 + e_i \} \tag{4}$$

where α_i $(i = 1,2,\cdots,N)$ are Lagrange multipliers. The KK-T conditions can be expressed by

$$\begin{cases} \frac{\partial L}{\partial w} = 0 & \Rightarrow \; w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \\ \frac{\partial L}{\partial b} = 0 & \Rightarrow \; \sum_{i=1}^{N} \alpha_i y_i = 0 \\ \frac{\partial L}{\partial e_i} = 0 & \Rightarrow \; \alpha_i = \lambda e_i \\ \frac{\partial L}{\partial \alpha_i} = 0 & \Rightarrow \; y_i [w^T \phi(x_i) + b] - 1 + e_i = 0 \end{cases} \quad i = 1,2,\cdots,N \tag{5}$$

Referring to Suykens and Gestel's work [5, 20], the solution of the optimization problem (2) can be obtained by solving the following linear equations:

$$\begin{bmatrix} 0 & y^{\mathrm{T}} \\ y & ZZ^T + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} \mathrm{b} \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \vec{1} \end{bmatrix} \tag{6}$$

where $Z = [\phi(x_1) y_1, \cdots, \phi(x_N) y_N]^T$, $y = [y_1, y_2, \cdots, y_N]^T$, $\vec{1} = [1,1,\cdots,1]^T$, $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_N]^T$ and $\Omega = ZZ^T$ takes the form as $\Omega_{kl} = y_k y_l \phi(x_k)^T \phi(x_l) = \psi(x_k, x_l)$ $(k,l = 1,2,\cdots,N)$ according to mercer's condition.

Table 1. Classical common kernel functions

Name	Function Expression
Linear Kernel	$\psi(x, y) = x^T y$
Polynomial Kernel	$\psi(x, y) = (1 + x^T y / \sigma^2)^d$
RBF Kernel	$\psi(x, y) = \exp\{-\lVert x - y \rVert_2^2 / \sigma^2\}$
SRBF Kernel	$\psi(x, y) = \exp\{-\sum_{i=1}^{n} (x_i - y_i)^2 / \sigma_i^2\}$
MLP* Kernel	$\psi(x, y) = \tanh(k x^T y + \theta)$

For the choice of the kernel function $\psi(\cdot,\cdot)$ one has several alternatives. Some of common kernel functions are listed in **Table 1.**, where c, d, σ, k and θ are constants, and for the function with "*" symbol. A suitable choice for k and θ is needed to enable the kernel function to satisfy Mercer condition.

After solving the Eq. (6), the LS-SVM model for classification can be obtained as:

$$y = \operatorname{sign}\left[\sum_{i=1}^{N} \alpha_i y_i \psi(x, x_i) + b\right] \tag{7}$$

3 PSO-Based Hyper-Parameters Selection for LS-SVM

3.1 Brief Introduction to PSO

The particle swarm optimization (PSO), originally developed by Kennedy and Elberhart [21], is a method for optimizing hard numerical functions on metaphor of social behaviors of flocks of birds and schools of fish. It is an evolutionary computation technique based on swarm intelligence. A swarm consists of individuals, called particles, which change their positions over time. Each particle represents a potential solution to the problem. In a PSO system, particles fly around in a multi-dimensional searching space. During its flight each particle adjusts its position according to its own experience and the experience of its neighboring particles, making use of the best position encountered by itself and its neighbors. The effect is that particles move towards the better solution areas, while still having the ability to search a wide area around the better solution areas. The performance of each particle is measured according to a pre-defined fitness function, which is related to the problem being solved. The PSO has been found to be robust and fast in solving non-linear, non-differentiable and multi-modal problems [22]. The mathematical description and executive steps of the PSO are as follows.

Let the i th particle in a D-dimensional space be represented as $\vec{x}_i = (x_{i1},\ldots,x_{id},\ldots,x_{iD})$. The best previous position of the i th particle is recorded and represented as $\vec{p}_i = (p_{i1},\ldots,p_{id},\ldots,p_{iD})$, which gives the best fitness value and is also called $pbest$. The index of the best $pbest$ among all the particles is represented by the symbol g. The location P_g is also called $gbest$. The velocity for the i th particle is represented as $\vec{v}_i = (v_{i1},\ldots,v_{id},\ldots,v_{iD})$. The concept of the particle swarm optimization consists of changing the velocity and location of each particle towards its $pbest$ and $gbest$ locations according to Eqs. (1) and (2) at each time step:

$$v_{id} = wv_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \,, \tag{8}$$

$$x_{id} = x_{id} + v_{id} \,, \tag{9}$$

where w is the inertia coefficient which is a constant in the interval [0, 1] and can be adjusted in the direction of linear decrease [23]; c1 and c_2 are learning rates which are nonnegative constants; r_1 and r_2 are generated randomly in the interval [0, 1]; $v_{id} \in [-v_{\max}, v_{\max}]$, and $v_{\max}$ is a designated maximum velocity. The termination criterion for iterations is determined according to whether the maximum generation or a designated value of the fitness is reached.

3.2 PSO-Based Hyper-Parameters Selection

There are two key factors to determine the optimized hyper-parameters using particle swarm optimization (PSO): one is how to represent the hyper-parameters as the particle's position, namely how to encode. Another is how to define the fitness function which evaluates the goodness of a particle. The following will give the two key factors.

Encoding Hype-parameters: The optimized hyper-parameters for LS-SVMs include kernel parameter(s) (except for linear kernel) and regularization parameter. In solving hyper-parameters selection by the PSO, each particle is requested to represent a potential solution, namely hyper-parameters combination. So let us denote an m hyper-parameters combination as a vector of dimension m. For example, SRBF: v=(γ , σ_1 , σ_2 , ..., σ_{ninput}), Pol: v=(γ , σ , d). The method of encoding is very intuitionistic. In this study, v=($\log\gamma$, $\log\sigma_1$, $\log\sigma_2$, ..., $\log\sigma_{ninput}$) and v=($\log\gamma, \log\sigma$, log$d$) is used because this gives a more stable optimization. For different kernels the length of the parameters vector is different.

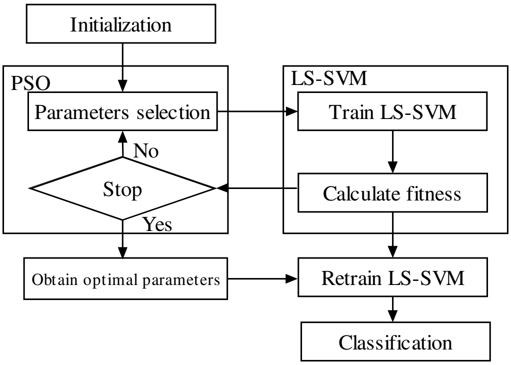


Fig. 1. Flow chart of PSO-based hyper-parameters algorithm

Fitness function: The fitness function is the generalization performance measure. For the generation performance measure, there are some different descriptions. Therefore the corresponding fitness can be determined. In this paper, the employed fitness function will be defined in Section 4.2.

The flow chart of the PSO-based hyper-parameters selection algorithm for the LS-SVM is shown in **Fig. 1**.

4 Numerical Experiments

4.1 Data sets and Its Preprocessing

Experiments are performed to evaluate the performance of PSOLS-SVM for binary classification. We selected the Diabetes (DB), Breast-Cancer (BC), Heart (HT), Thyroid (TD) and Titanic (TC) data sets from the UCI Machine Learning repository. The data sets used in this study are provided by G. Ratsch at http://ida.first.gmd.de/aetsch/data/benchmarks.htm. The detailed description of these data sets is reported in **Table 2**. These data sets have been referred to numerous times in the literature, which makes them very suitable for benchmarking purposes. The data are preprocessed and partitioned as in [24]: Each component of the input data is normalized to zero mean and unit standard deviation. It ensures the larger value input attributes do not overwhelm smaller value inputs; hence helps to reduce errors. After normalized to zero mean and unit standard deviation, each data set is divided randomly 100 times into different pairs of disjoint train and test sets.

Table 2. Description of the data sets

	DB	BC	HT	TD	TC
N_{train}	468	200	170	140	150
N_{test}	300	77	100	75	2051
N	768	277	270	215	2201
n_{input}	8	9	13	5	3

N_{train} and N_{test} denotes the number of train and test patterns, respectively. N stands for the total number of the patterns. n_{input} is the number of the input.

4.2 Determination of the Fitness Function

In PSO, the fitness value is used to evaluate goodness of the particles, namely hyper-parameter combination. So the determination of fitness function is important to the parameters of LS-SVM. The fitness should reflect the generalization performance of LS-SVM for different hyper-parameter combination. The fitness function is defined as follows: For each particle, five LS-SVMs are built using the training sets of the first five data partitions and the average of the classification correct rates on the corresponding five test sets determines the fitness value (training performance of LS-SVM). The particle with the largest fitness is chosen as the optimal parameters combination [25]. The test performance of LS-SVM with optimal parameters is measured as follows: 100 LS-SVMs are built using the optimal parameters using all the training sets and the average of the classification correct rates on the corresponding 100 test sets is define as the test performance of LS-SVM.

4.3 Experiment Results

All experiments are performed on a PC with Pentium IV 2.6GHz processor and 512MB memory. The optimal values for the regularization parameter and the kernel parameters with linear, polynomial, RBF and SRBF kernel are shown in **Table 3**. The first column is the parameters used with different kernels. The rest column is the optimal parameters values for different data sets. For polynomial kernel, the degree is denoted in bracket. The corresponding optimal values are not given because of the large number of parameters. In **Table 4** and **Table 5**, the first column lists the different kernels and the first row shows the benchmark data sets in our study, respectively. **Table 4** and **Table 5** show the performance of training and test of LS-SVM on different data sets, respectively. Experimental results in **Table 5** show that the SRBF kernel yields the best test performance and the polynomial and RBF kernel give good test performance. **Table 6**, in which test error found by PSO-based hyper-parameters selection of LS-SVM and other methods for different data sets is listed, shows that the results obtained from the proposed method for LS-SVM with SRBF kernel are better than those in literature [24].

Table 3. Optimized hyper-parameter values of the LS-SVM with linear, RBF and polynomial kernels for different data sets

	BC	HT	TiD	DS	TC
Lin: $\log_{10}(\gamma)$	-0.23	-2.26	0.68	-1.06	-1.67
Pol: $\log_{10}(\gamma)$	1.35	1.36	1.30	1.08	1.69
Pol: $\log_{10}(\sigma)$	1.39	2.16	0.62	1.34	-0.23
Pol: $\log_{10}(d)$	0.76 (5)	0.71 (5)	0.83 (6)	0.75 (5)	0.68 (4)
RBF: $\log_{10}(\gamma)$	0.33	1.41	1.27	1.92	4.00
RBF: $\log_{10}(\sigma)$	0.76	1.83	0.26	1.20	0.58

Table 4. LS-SVM training performance with different kernel functions by using the optimized parameters for different data sets

	BC	HT	TD	DS	TC
Lin	72.99	83.00	84.53	76.73	77.59
Pol	74.81	83.00	93.33	77.20	78.43
RBF	74.81	83.00	97.07	77.27	77.60
SRBF	79.74	86.20	98.40	78.00	77.55

Table 5. LS-SVM test performance with different kernel functions by using the optimized parameters for different data sets

	BC	HT	TD	DS	TC
Lin	72.98	84.41	85.21	76.63	77.33
Pol	73.66	84.41	92.01	77.01	77.02
RBF	73.82	84.42	95.99	77.05	78.10
SRBF	76.09	84.49	96.59	77.46	78.41

Table 6. Test performance found by PSO-based hyper-parameters selection of LS-SVM and other methods for different data sets

	BC	HT	TD	DS	TC
RBF-Network	72.36	82.45	95.48	76.71	76.74
AdaBoost with RBF-Network	69.64	79.71	95.60	73.53	77.42
LP_Reg-AdaBoost	73.21	82.51	95.41	75.89	76.02
QP_Reg-AdaBoost	74.09	82.83	95.65	74.61	77.29
AdaBoost_Reg	73.49	83.53	95.45	76.21	77.36
SVM with RBF-Kernel	73.96	84.05	95.20	76.47	77.58
KFD with RBF-Kernel	75.23	83.86	95.80	76.79	76.75
LS-SVM with SRBF-Kernel	**76.09**	**84.49**	**96.59**	**77.46**	**78.41**

5 Conclusions

A promising novel particle swarm optimization-based hyper-parameters selection for LS-SVM classifier is proposed. The presented method does not consider the analytic property of the generalization performance measure and the number of hyper-parameters. The feasibility of our presented method is evaluated on benchmark data sets. Experimental results show that better performance can be obtained. Experiments on SRBF kernel show that the proposed method can tune much more hyper-parameters. Experimental results also show that the SRBF kernel yields the best test performance and the polynomial and RBF kernel gives better test performance. Compared with the results of other methods, the proposed PSO-based hyper-parameters selection for LS-SVM yields higher accurate rate for all data sets tested in this paper.

Acknowledgment

The authors are grateful to the support of the National Natural Science Foundation of China (60433020), the science-technology development project of Jilin Province of China (20050705-2), the doctoral funds of the National Education Ministry of China (20030183060), the graduate innovation lab of Jilin University (503043), and “985”project of Jilin University.

References

1. Vapnik V. N.: Statistical Learning Theory. John Wiley, New York, USA (1998)
2. Christianimi N., Shawe-Taylor J.: An Introduction to Support Vector Machines and other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)
3. Gunn S. R.: Support Vector Machines for Classification and Regression. Technical Report. University of Southampton (1998)
4. Kim K. J.: Financial Time Series Forecasting Using Support Vector Machines. Neurocomputing. 1-2 (2003) 307-319

5. Suykens J. A. K, Vandewalle J.: Least Squares Support Vector Machine Classifiers. Neural Processing Letters. 9 (1999) 293-300
6. Chapelle O., Vapnik V. N., Bousquet O., Mukherjee S.: Choosing Multiple Parameters for Support Vector Machines. Machine Learning. 1-3 (2002) 131-159
7. Vapnik V. N., Chapelle O.: Bounds on Error Expectation for Support Vector Machines. Neural computation. 9 (2000) 2013-2036
8. Chapelle O., Vapnik V. N.: Model Selection for Support Vector Machines. In: Solla S., Leen T., Müller K.-R. (Eds.): Advances in Neural Information Processing Systems 12: Proceedings of the 1999 Conference, Vol. 12. MIT Press, Cambridge, MA (2000) 230-236
9. Chung K. -M., Kao W. -C., Sun C. -L., Lin C.-J.: Radius Margin Bound for Support Vector Machines with RBF Kernel. Neural Computation. 11 (2003) 2643-2681
10. Wahba G., Lin X., Gao F., Xiang D., Klein R., Klein B.: The Bias-variance Trade-off and the Randomized gacv, In: Kearns M., Solla S., Cohn D. (Eds.): Advances in Neural Information Processing Systems 11: Proceedings of the 1998 Conference. Vol. 11, MIT Press, Cambridge, MA (1999) 620–626
11. Sathiya Keerthi S.: Efficient Tuning of SVM Hyperparameters Using Radius/Margin Bound and Iterative Algorithms. IEEE Transactions on Neural Network. 5 (2002) 1225-1229
12. Ayat N. E., Cheriet M., Suen C. Y.: Automatic Model Selection for the Optimization of SVM Kernels. Pattern Recognition. 10 (2005) 1733-1745
13. Gold C., Sollich P.: Model Selection for Support Vector Classification. Neurocomputing. 1-2 (2003) 221-249
14. Eads D. R., Hill D., Davis S., Perkins S. J., Ma J., Porter R. B., Theiler J. P.. Genetic Algorithms and Support Vector Machines for Time Series Classification. In: Bosacchi B., Fogel D. B., Bezdek J. C. (Eds.): Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation V. Proceedingsof the SPIE, Vol. 4787. (2002) 74–85
15. Frohlich H., Chapelle O., Scholkopf B.: Feature Selection for Support Vector Machines by Means of Genetic Algorithms. In: Proceedings of the 15th IEEE International Conference on Tools with AI (ICTAI 2003). IEEE Computer Society. Institute of Electrical and Electronics Engineers Inc., USA (2003) 142–148
16. Jong K., Marchiori E., r Vaart A. van de: Analysis of Proteomic Pattern Data for Cancer Detection. In: Raidl G. R., Cagnoni S., Branke J., Corne D. W., Drechsler R., Jin Y., Johnson C. G., Machado P., Marchiori E., Rothlauf F., Smith, G. D. Squillero G. (Eds.): Applications of Evolutionary Computing. Lecture Notes in Computer Science, Vol. 3005, Springer Berlin, Heidelberg (2004) 41–51
17. Miller M. T., Jerebko A. K., Malley J. D., Summers R. M.: In: Clough A. V., Amini A. A. (Eds.): Feature Selection for Computer-aided Polyp Detection Using Genetic Algorithms, Medical Imaging 2003: Physiology and Function: Methods, Systems, and Applications, Proceedings of the SPIE, Vol. 5031, (2003) 102–110.
18. Ping-Feng Pai, Wei-Chiang Hong: Support Vector Machines with Simulated Annealing Algorithms in Electricity Load Forecasting. Energy Conversion & Management. 17 (2005) 2669-2688
19. Frauke Friedrichs, Christian Igel: Evolutionary Tuning of Multiple SVM Parameters. Neurocomputing. 64 (2005) 107-117
20. Gestel T. V., Suykens J. A. K., Baesens B., Viaene S., Vanthienen J., Dedene G., Moor B. D., Vandewalle J.: Benchmarking Least Squares Support Vector Machine classifiers. Machine Learning. 1 (2004) 5-32

21. Kennedy J., Eberhart R.: Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia. IEEE Service Center, Vol. 4. Piscataway, NJ (1995) 1942–1948
22. Ge H. W., Liang Y. C., Zhou Y., Guo X. C.: A Particle Swarm Optimization-based Algorithm for Job-shop Scheduling Problem. International Journal of Computational Methods. 3 (2005) 419-430
23. Shi Y., Eberhart R.: A Modified Particle Swarm Optimizer. IEEE World Congress on Computational Intelligence, Alaska, ALTEC, Vol. 1 (1998) 69-73
24. Ratsch G., Onoda T., Muller K.-R.: Soft Margins for Adaboost. Machine Learning. 3 (2001) 287–330
25. Meinicke P., Twellmann T., Ritter H.: Discriminative Densities from Maximum Contrast Estimation. In: Becker S., Thrun S., Obermayer K. (Eds.): Advances in Neural Information Processing Systems, Vol. 15. MIT Press, Cambridge (2002) 985–992

Training RBF Neural Networks with PSO and Improved Subtractive Clustering Algorithms

JunYing Chen[1] and Zheng Qin[1,2]

[1] Department of Computer Science, Xian JiaoTong University,
Xian 710049, P.R. China
[2] School of Software, Tsinghua University, Beijing 100084, P.R. China
vcjy@163.com

Abstract. In this paper, Particle Swarm Optimization (PSO) and improved subtractive clustering algorithm were proposed for training RBF neural networks. PSO was used to feature selection in conjunction with RBF classifiers for individual fitness evaluation. During RBF training process, supervised mean subtractive clustering algorithm (SMSCA) was used to evolve RBF networks dynamically with the selected feature subset based on PSO algorithm. Experimental results on four datasets show that RBF networks evolved by our proposed algorithm have more simple architecture and stronger generalization ability with nearly the same classification performance when compared with the networks evolved by other methods.

Keywords: Particle Swarm Optimization, Subtractive Clustering Algorithm, RBF Neural Network, Feature Selection.

1 Introduction

Both the dimensionality and the distribution of input patterns affect the number of radical basis functions in RBF networks. Reducing the dimensionality and representing the distribution of the input patterns are two critical ways to simplify the architecture and improve classification accuracy of RBF neural networks.

In a large number of features measured in pattern recognition applications, there are always some irrelevant features. If no preprocessing is carried out before patterns are used to train RBF neural networks, the size of RBF neural networks may be too large. In order to simplify the architecture of RBF neural networks, dimensionality reduction techniques are often useful. Feature selection, a kind of dimensionality reduction technique, aims to select the best subset of features out of the original set. Feature selection can reduce the computational cost of feature measurement, simplify the architecture of classifiers, and increase the classification accuracy [1, 2].

Clustering algorithms are able to find cluster centers best representing data distribution. Hence clustering algorithms have been successfully used in training RBF neural networks. In [3] the optimal partition algorithm (OPA) was used to determine the centers and widths of radial basis functions for time series forecasting. In [4] clustering algorithm using a mixed possibilistic and fuzzy approach was proposed to

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1148 – 1155, 2006.

evolve the center vector of the hidden layer. In [5] the forward selective clustering with cluster sample transform algorithm determined the initial number and center vectors of hidden units. The research in [6] compared the performance of the RBF neural networks evolved by seven different clustering techniques. In most traditional algorithms, such as the K-means, the number of cluster centers need to be predetermined, which restricts the real applications of the algorithms.

In this paper, feature selection was carried out in order to reduce the number of attributes as well as the complexity of RBF neural networks. A PSO algorithm was used to reduce the number of irrelevant attributes using a RBF classifier as an individual's fitness measure. Supervised mean subtractive clustering algorithm (SMSCA) was proposed to evolve RBF networks with the selected feature set. There is no need to predetermine the number of cluster centers in SMSCA. The results of computational experiments showed SMSCA can improve the performance of RBF neural networks and feature selection can reduce feature size and simplify RBF neural networks.

The rest of the paper is organized as follows. Section 2 presents the architecture of the RBF networks used. Section 3 describes the PSO learning algorithm for feature selection. In Section 4, SMSCA for evolving RBF networks are discussed. Parameter selection and experimental results are presented in Section 5. Finally, the conclusions are given in Section 6.

2 Radial Basis Function Neural Network

Radial Basis Function neural network (RBFNN) is a kind of feed-forward neural network. RBFNN has been widely used in many pattern recognition applications because of its simple architecture and easily learning ability.

RBFNN is a three-layer network. The input layer consists of n units which transfer the input to the hidden layer. The hidden layer is composed of some basis functions that execute non-linear mapping. The output of a neuron in the output layer can be achieved by computing the weighted sum of outputs of hidden layer. The RBFNN form with linear combination of Gaussian functions is shown in the following.

$$o_i(x) = \sum_{k=1}^{N} w_{ik} \exp\{-\frac{\left\| x - c_k \right\|^2}{2\sigma_k^2}\}, i = 1, 2, ..., m \ . \qquad (1)$$

where $|| \dots ||$ represents Euclidean norm, c_k , σ_k and w_{ik} are the center, the width of the k-th neuron in the hidden layer and the weights in the output layer respectively, m is the number of neurons in the output layer. N is the number of neurons in the hidden layer.

One kind of training methods regarding RBF networks is clustering. The centers and widths of RBF neural networks are determined based on clustering method. The weights between the hidden layer and the output layer are computed by solving linear equations. The clustering method is crucial to the performance of RBF neural networks.

3 Feature Selection with Particle Swarm Optimization

PSO is a new population-based evolutionary computation technique firstly proposed in 1995 [7]. Particle swarms explore the search space through a population of particles. The particle evolves by adjusting its position at diverse speed iteratively. The position of the *i*-th particle at *t* iteration is represented by $X_i^{(t)}=(x_{i1}, x_{i2},,, x_{iD})$, and its velocity is represented by $V_i^{(t)}=(v_{i1}, v_{i2},,, v_{iD})$. The movement of the particle is not only influenced by the particle's own memories but also the memories of its neighborhood. The position with the best fitness value visited by the *i*-th particle is donated by P_i and the position with the best fitness found by all particles is donated by P_g . The velocity update equation (2) and position update equation (3) are described as follows:

$$V_i^{(t+1)} = w*V_i^t + c_1 * rand()*(P_i - X_i^{(t)}) + c_2 * rand()*(P_g - X_i^{(t)}) \ . \tag{2}$$

$$X_i^{(t+1)} = X_i^{(t)} + V_i^{(t)} \ . \tag{3}$$

where w is inertia weight which balances the global exploitation and local exploration abilities of the particles, c_1 and c_2 are acceleration constants, *rand()* are random values between 0 and 1. The velocities of the particles are limited in $[Vmin, Vmax]^D$. If smaller than *Vmin*, an element of the velocity is set equal to *Vmin*, if greater than *Vmax*, and then set equal to *Vmax*.

When PSO is used to feature selection, there are two important problems, encoding particles and designing fitness function. Each particle is encoded as a combination of variables, which represent the relevant information of feature subset that needs to be determined. Each particle is a potential solution to the feature selection problem. The fitness function gives directions to particle swarm. Through searching in the variable space, particle swarm finds the final solution considered to be the solution to the problem.

Encoding Particles. When encoding particles, a principal problem lies in representing all possible feature subsets. As shown in Equation 4, each particle is encoded into a real-value vector to perform selection of a subset of the features.

$$[(f_1, f_2, ..., f_n)] \ . \tag{4}$$

where n is the number of features. $f_i (i = 1, 2, ..., n)$ represents whether or not the *i*-th feature is selected. If the value for a given feature is negative, the feature is not considered for classification. If the corresponding value is positive, the feature is included in the classifier.

Fitness Evaluation. Fitness evaluation function involves the output error of RBF network and the size of feature subset selected. The training method of RBF networks is introduced in next part 4. To minimize the output error and the size of RBF networks, the fitness function was formulated as equation (5).

$$E = N_t \times \log(\frac{1}{N_t}\sum_{n=1}^{N_t} \|y_n - o_n\|^2) + \lambda \times N_f \ . \tag{5}$$

where N_t is the number of training patterns, y_n, o_n are the desired output and network output for pattern n respectively, N_f is the number of features involved in networks, λ is a parameter that balances classification performance and the size of the selected feature set.

Feature Selection Algorithm. The algorithm of feature selection with PSO is presented in the following:

1. Initialize swarm of *N* particles. Each particle defines the information of a feature subset. Set the number of iterations as *MaxIteration*. Set *count*=0.

2. Decode each particle and execute selecting operations on patterns to form new patterns. Compute the fitness of each particle with equation (5) based on the new patterns.

3. Update P_i for each particle and P_g for whole swarm.

4. Update the velocity of each particle according to formula (2). Limit the velocity in $[Vmin, Vmax]^D$.

5. Update the position according to formula (3).

6. Set *count*=*count*+1; if *count* < *MaxIteration*, go to 2; otherwise, Terminate the algorithm.

4 SMSCA for Evolving RBF Neural Networks

RBF neural networks have been widely used for function approximation, pattern classification and time series prediction and so on. In this paper, we used RBF classifier as fitness evaluating criteria to select the appropriate subset of feature set. Based on the feature subset, the RBF neural network with good performance was evolved simultaneously. Hence it was important to determine RBF neural network quickly.

4.1 Supervised Mean Subtractive Clustering Algorithm

Subtractive clustering algorithm (SCA) [8] is an unsupervised learning method based only on input training patterns. It obtains cluster centers by selecting the data point with the highest potential value iteratively. The basic SCA selects cluster centers only from the input patterns, but the cluster centers best representing data distribution are not necessarily in the original input patterns. In our proposed SMSCA, mean methods were used to derive cluster centers from a high-density area around high potential data point. In additional, the classification information was also included to compute the potential value of the input pattern.

Given n input patterns $(x_1, x_2, \cdots x_n)$ in d-dimensional space, SMSCA can be described as follows:

Step1: set the radii of the high-density area r_a and the number of cluster centers $k=1$. Initialize the potential value $P_i = 0$ ($i = 1, 2, \ldots, n$).

Step 2: for every input pattern x_j, compute the potential of x_i to serve as a cluster center as P_i by equation (6) if x_i and x_j are in the same class.

$$P_i = P_i + \exp[-\frac{\left\| x_i - x_j \right\|^2}{(r_a/2)^2}] \quad j = 1,2,\ldots,n \ . \tag{6}$$

Step3: select the input pattern with the highest potential as c_k. Compute the location of the cluster center $\overline{c_k}$ by equation (7) and its potential value $\overline{P_k}$ by equation (6) only substituting x_i with $\overline{c_k}$.

$$\overline{c_k} = \frac{1}{m}\sum_{j=1}^{m} x_j^{(k)} \quad . \tag{7}$$

where $(x_1^{(i)}, x_2^{(i)}, \cdots x_m^{(i)})$ is the input patterns with the same class of c_i that locate in the neighborhood of c_i defined by r_a.

Step4: Subtract the potential of each input pattern x_i in the same class of $\overline{c_k}$ by equation (8).

$$P_i = P_i - \overline{P_k}\, e^{(-\frac{\left\| x_i - \overline{c_k} \right\|^2}{(r_b/2)^2})} \quad . \tag{8}$$

where r_b is a positive radius defining the neighborhood in which the input patterns reduce potential value greatly and will unlikely be selected as the next cluster center. To avoid obtaining closely spaced cluster centers, $r_b = 1.5 r_a$ was chosen.

Step5: if $\overline{P_k} < 0.15 * \overline{P_1}$, terminate; else k=k+1 and return step 3.

4.2 Evolving RBF Neural Networks

Once the cluster centers have been fixed by SMSCA, each cluster center was taken as the center vector of a radial basis function. The width of k-th radial basis function was computed by the following equation (9).

$$\sigma_k = \frac{1}{N-1}\sum_{i=1, i \neq k}^{N} \left\| c_k - c_i \right\| \ . \tag{9}$$

The optimal output weights were determined by pseudo-inverse algorithm without having local minima problem.

The value of r_a influences the number and the locations of cluster centers. The larger r_a is, the fewer number of centers is. The expected classification accuracy and network size can be obtained by adjusting r_a. During training RBF as a fitness evaluation function, the larger r_a may be used to reduce the computation cost. When feature selection algorithm terminates, the smaller r_a is set to evolve RBF with the selected feature subset over again.

5. Experiments

5.1 Experimental Setup

The parameters of the PSO algorithm were set as follows: weight *w* decreasing linearly between 0.9 and 0.4, learning rate $c_1 = c_2 = 2$ for all cases. The population size used by PSO was constant. The algorithm stopped when a predefined number of iterations have been reached. Once finished the long set of experiments, values selected for parameters were shown in tables 1.

Table 1. Execution parameters for feature selection algorithm with PSO

Parameter	value
Population Size	20
Iterations	1000
Vmax	2
λ	4

Before using SMSCA, input feature values must be normalized over the range [0, 1] as follows:

$$x_{i,j}^{'} = \frac{x_{i,j} - \min\limits_{k=1,\ldots,n}(x_{k,j})}{\max\limits_{k=1,\ldots,n}(x_{k,j}) - \min\limits_{k=1,\ldots,n}(x_{k,j})} \quad . \tag{9}$$

Where $x_{i,j}$ is the *j*th feature of the *i*th pattern, $x_{i,j}^{'}$ is the corresponding normalized feature, and *n* is the total number of patterns.

5.2 Experimental Results

Evaluations of feature selection and clustering techniques on RBF training tasks were developed by using four well-known real databases, wine, thyroid, ionosphere and wdbc [9]. For comparison, three training schemes were considered. One was SCA for RBF training based on full feature set, denoted by SCA-RBF, another was SMSCA for RBF training based on full feature set, denoted by SMSCA -RBF and the other was SMSCA-RBF algorithm with feature selection used in this paper. r_a was set the same value for three training schemes on the same dataset, 0.6, 0.1, 0.28 and 0.36 for wine, thyroid, ionosphere and wdbc database respectively. Once the feature subset was obtained, the smaller r_a was set to evolve RBF networks once again.

Each benchmark was tested with 5-fold cross-validation except SMSCA-RBF in feature selection was tested with 3-fold cross-validation for fitness evaluation. The results were listed in Table 2. Train Accuracy and Test Accuracy referred to mean

Table 2. The basic information of datasets, the results achieved by RBF networks evolved by three training algorithms respectively

Dataset	wine	thyroid	ionosphere	wdbc
The basic information of datasets				
Instances	178	215	351	569
Features	13	5	34	30
Classes	3	3	2	2
The results achieved by SCA-RBF				
Train Accuracy	0.9883	0.9667	0.9346	0.9631
Test Accuracy	0.9743	0.9409	0.9083	0.9581
Hidden units	16	18	38	17
The results achieved by SMSCA-RBF				
Train Accuracy	0.9900	0.9731	0.9434	0.9568
Test Accuracy	0.9789	0.9493	0.9191	0.9527
Hidden units	13	16	37	11
The results achieved by SMSCA-RBF with feature selection				
Selected features	5	3	8	9
Train Accuracy	0.9784	0.9729	0.9399	0.9624
Test Accuracy	0.9714	0.9619	0.9289	0.9607
Hidden units	9	13	16	9

correct classification rate averaged over 10 runs for the training and testing set, respectively. The number of selected features and hidden units achieved for each dataset were averaged over 10 runs, and then were round off to integer.

By comparing the results, it can be seen that SMSCA-RBF outperformed SCA-RBF in both classification accuracy and network size on wine, thyroid and ionosphere dataset. This is probably because SMSCA can find cluster centers representing data distribution better than SCA does. SMSCA-RBF with feature selection algorithm found the small-sized feature sets. And the networks evolved by the proposed algorithm based on the found small feature sets have less hidden units than those evolved by SMSCA-RBF based on the original feature sets. The generalization ability of the RBF neural networks improved on thyroid, ionosphere and wdbc dataset. In our proposed method, classification accuracy still remains high in despite of many features removed from the original feature set.

6 Conclusions

In this paper, supervised mean subtractive clustering algorithm was proposed to evolve RBF neural networks and the evolved RBF acts as fitness evaluation function of PSO algorithm for feature selection. The method performs feature selection and RBF training simultaneously. The appropriate feature subset was selected according to the performance of RBF networks, and the RBF network with good performance was achieved based on the selected feature subset. Experimental results show that the proposed methods are effective in reducing the feature size, the structural complexity of the RBF neural network, and even the classification error rates.

Acknowledgments. This work was supported by the national grand fundamental research 973 program of China (No.2004CB719401).

References

1. Fu, X.J., Wang, L.: Data dimensionality reduction with application to simplifying RBF network structure and improving classification performance. IEEE Transactions on Systems Man and Cybernetics, 33(3) (2003) 399 – 409
2. Scherf, M., Brauer, W.: Improving RBF networks by the feature selection approach EUBAFES. In: Proceedings of the 7th International Conference on Artificial Neural Networks, Vol. 1327. Springer-Verlag, Berlin Heidelberg Lausanne (1997)391-396
3. Sun, Y.F., Liang, Y.C., Zhang W.L., etc.: Optimal partition algorithm of the RBF neural network and its application to financial time series forecasting. Neural Computing and Applications,14(1) (2005)36-44
4. Guilen, A., Rojas, I., Gonzalez, J., etc.: Possibilistic Approach to RBFN Centers Initialization. International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, 2 (2005)174-183
5. Sun, J., Shen, R.M., Yang, F.: An Adaptive Learning Algorithm Aimed at Improving RBF Network Generalization Ability. Lecture Notes in Artificial Intelligence, Vol. 2557. Springer-Verlag, Berlin Heidelberg Canberra, Australia(2002)
6. Carvalho, D., Brizzotti, M.M.: Combining RBF Networks Trained by Different Clustering Techniques. Neural Processing Letters, 14(3) (2001) 227-240
7. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ (1995) 1942-1948
8. Chiu, S.: Fuzzy model identification based on cluster Estimation (J). Journal of Intelligent and Fuzzy Systems, (2)3 (1994)209-219.
9. Blake, C., Merz, C.J.: UCI Repository of Machine Learning Databases. http://www.ics.uci.edu/~mlearn/MLRepository.html. (1998)

Training RBF Neural Network Via Quantum-Behaved Particle Swarm Optimization

Jun Sun, Wenbo Xu, and Jing Liu

Center of Intelligent and High Performance Computing,
School of Information Technology, Southern Yangtze University
No. 1800, Lihudadao Road, Wuxi,
214122, Jiangsu, China

Abstract. Radial Basis Function (RBF) networks are widely applied in function approximation, system identification, chaotic time series forecasting, etc. To use a RBF network, a training algorithm is absolutely necessary for determining the network parameters. The existing training algorithms, such as Orthogonal Least Squares (OLS) algorithm, clustering and gradient descent algorithm, have their own shortcomings. In this paper, we make an attempt to explore the applicability of Quantum-behaved Particle Swarm Optimization, a newly proposed evolutionary search technique, in training RBF neural network. The proposed QPSO-Trained RBF network was test on nonlinear system identification problem, and the results show that it can identifying the system more quickly and precisely than that trained by Particle Swarm algorithm.

1 Introduction

Radial Basis Functions, as a variant of feed-forward artificial neural network, have been successfully applied to a large diversity of applications including interpolation [2], chaotic time-series modeling [3], system identification [11], etc. In order to use a Radial Basis Function network we need to specify the hidden unit activation function, the number of processing units, a criterion for modeling a given task and, in turn, a training algorithm for finding the parameters of network. Finding the RBF weight is called network training. The most widely used training algorithms for RBF network include Orthogonal Least Squares (OLS) algorithm, clustering and gradient-based algorithm, etc ([2], [4], [7], [15], [11], [18]). These algorithms, however, possess their shortcomings. Evolutionary algorithms are a class of population-based search techniques, which have strong global search ability and robustness and could be used to training RBF and other neural networks, and become promising training algorithms for neural networks.

Recently, a novel evolutionary technique, Quantum-behaved Particle Swarm Optimization (QPSO), has been proposed ([12], [13], [14]). It has been shown that QPSO outperforms original Particle Swarm Optimization (PSO) considerably on several widely known benchmark functions. In this paper, we will explore the applicability of QPSO in training RBF neural network. The paper is structured as follows. In Section 2, RBF network model and parameter selection problem are

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1156 – 1163, 2006.

introduced. Section 3 describes QPSO algorithm. In Section 4, we propose our QPSO-Trained RBF network model. Section 5 gives the experiments results of the proposed model on system identification problem. Finally, the paper is concluded in Section 6.

2 Structure and Parameter Selection of RBF Neural Network

RBF Neural Network is structured by embedding radial basis function a two-layer feed-forward neural network. Such a network is characterized by a set of inputs and a set of outputs. In between the inputs and outputs there is a layer of processing units called hidden units. Each of them implements a radial basis function. Mathematically the RBF network can be formulated as

$$g(x)=\sum_{k=1}^{m}\lambda_k\varphi_k(\|x-c_k\|) \tag{1}$$

where m is the neuron number of hidden layer, which is equal to cluster number of training set. $\|x-c_k\|$ stands for the distance between the data point x and the RBF center c_k. λ_k is the weight related with RBF center c_k. Therefore, the RBF neural networks output is a weighted sum of the hidden layer's activation functions. In this paper, we adopt the most commonly used Gaussian RB functions as basis (activation) functions, then in the formula (1),

$$\varphi_k(x)=R_k(x)\Big/\sum_{i=1}^{m}R_i(x)\text{,}\quad R_k(x)=\exp\left(-\frac{\|x-c_k\|^2}{2\sigma_k^2}\right) \tag{2}$$

In formula (2), σ_k indicates the width of the kth Gaussian RB functions. One of the σ_k selection methods is shown as follows.

$$\sigma_k^2=\frac{1}{M_k}\sum_{x\in\theta_k}\|x-c_k\|^2 \tag{3}$$

where θ_k is the kth cluster of training set and M_k is the number of sample data in the kth cluster.

The neuron number of the hidden layer, i.e., the cluster number of training set, must be determined before the parameter selection of RBF neural network. In this paper, we adopt an efficient method, Rival Penalized Competitive Learning (RPCL) [17], to decide the cluster number. If the neuron numbers of hidden layer has been decided, the performance of RBF depends on the selection of the network parameters. There are three types of parameters in a RBF neural network model with Gaussian basis functions: (1). RBF centers (hidden layer neurons); (2). Widths of RBFs (standard deviations in the case of a Gaussian RBF); (3). Output layer weights. Different strategies exist for training of RBF neural network models. By means of training, the neural network models the underlying function of a certain mapping. In order to model such a mapping we have to find the network weights and topology. There are two categories of training algorithms: supervised and unsupervised. RBF networks are used mainly in supervised applications. In a supervised application, we are provided with a set of data samples called training set for which the corresponding network outputs are known. In this case the network parameters are found such that they minimize a cost function. In unsupervised training the output assignment is not available for the given set.

3 Quantum-Behaved Particle Swarm

Particle Swarm Optimization (PSO) algorithm, originally introduced by Kennedy and Eberhart in 1995 [9], simulates the knowledge evolvement of a social organism, in which each individual is treated as an infinitesimal particle in the *n*-dimensional space, with the position vector and velocity vector of particle *i* being represented as $X_i(t) = (X_{i1}(t), X_{i2}(t), \cdots, X_{in}(t))$ and $V_i(t) = (V_{i1}(t), V_{i2}(t), \cdots, V_{in}(t))$. The particles move according to the following equations:

$$V_{ij}(t+1) = w \cdot V_{ij}(t) + c_1 \cdot r_1 \cdot (P_{ij}(t) - X_{ij}(t)) + c_2 \cdot r_2 \cdot (P_{gj}(t) X_{ij}(t)) \tag{4}$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1) \quad i = 1,2,\cdots M\ ;\ j = 1,2\cdots,n \tag{5}$$

where c_1 and c_2 are called the acceleration coefficients. Vector $P_i = (P_{i1}, P_{i2}, \cdots, P_{in})$ is the best previous position (the position giving the best fitness value) of particle *i* known as the *personal best position* (pbest); vector $P_g = (P_{g1}, P_{g2}, \cdots, P_{gn})$ is the position of the best particle among all the particles in the population and is known as the *global best position* (gbest). The parameters r_1 and r_2 are two random numbers distributed uniformly in (0,1). Generally, the value of V_{ij} is restricted in the interval $[-V_{\max}, V_{\max}]$. Inertia weight *w* was first introduced by Shi and Eberhart in order to accelerate the convergence speed of the algorithm [16].

Trajectory analyses in [6] demonstrated the fact that convergence of the PSO algorithm may be achieved if each particle converges to its local attractor with coordinates

$$p_{ij}(t) = \varphi \cdot P_{ij}(t) + (1-\varphi) \cdot P_{gj}(t), where \quad \varphi = c_1 r_1 / (c_1 r_1 + c_2 r_2) \tag{6}$$

In QPSO, each individual quantum-behaved particle moves in a search space with each dimension existing a Delta Potential Well, whose center is p_{ij}. We can get the following update equation for position of the particle [12].

$$X_{ij} = p_{ij} \pm \frac{L}{2} \ln(1/u) \tag{7}$$

The value of *L* and the position are evaluated by $L = 2\alpha \cdot |C_j(t) - X_{ij}(t)|$, where C_j is defined as the mean of the *personal best* positions among all particles, that is $C_j(t) = \frac{1}{M} \sum_{i=1}^{M} P_{ij}(t)$. Thus we can get the following iterative equation for QPSO [13].

$$X_{ij}(t+1) = p_{ij}(t) \pm \alpha \cdot \left|C_j(t) - X_{ij}(t)\right| \cdot \ln(1/u) \tag{8}$$

where parameter α is called Contraction-Expansion (CE) Coefficient, which can be tuned to control the convergence speed of the algorithms. For more detailed information of QPSO, one may see literatures such as [12], [13] and [14].

4 QPSO-Trained RBF Neural Network

When training RBF NN by QPSO, a decision vector represents a particular group of network parameters including c_k, λ_k and c_k $(k = 1,2,\cdots,m)$. Thus each particle flies in a 3m-dimensional search space with $X_i = (c_1, c_2, \cdots, c_m, \sigma_1, \sigma_2, \cdots, \sigma_m, \lambda_1, \lambda_2, \cdots, \lambda_m)$ denoting its position. Initialization of the population involves generating randomly the position vector X_i $(i = 1,2,\cdots,M)$ and setting the personal best position $P_i = X_i (i = 1,2,\cdots,M)$.

Since a component of the position corresponds to a network parameter, a RBF network is structured according the particle's position vector. Training the corresponding network by inputting the training samples, we can obtain an error value computed by the following formula.

$$E = \frac{1}{2Q}\sum_{j=1}^{Q}\sum_{s=0}^{c}[y_{s,j}(x_j) - g_{s,j}(x_j)]^2 \tag{9}$$

where $y_{s,j}(x_j)$ and $g_{s,j}(x_j)$ are the actual response (output) and network's predicted response (output) at output unit s on x_j, respectively. Q is the number of the training sample and c is the number of output units. The particle is evaluated by the obtained error value (fitness value), by which it can be determined whether P_i and P_g need to be updated. In a word, the error function (9) is adopted as the objective function to be minimized in QPSO-based RBF neural network.

There are two alternatives for stop criterion of the algorithm. One method is that the algorithm stops when the objective function value is less than a given threshold ε; the other is that it terminates after executing a pre-specified number of iterations. The following is the procedure of QPSO-Trained RBF neural network algorithm:

(1) Initialize the population by randomly generate the position vector X_i of each particle and set $P_i = X_i$;

(2) Structure a RBF neural network by treating the position vector of each particle as a group of network parameter;

(3) Training each RBF network on the training set;

(4) Evaluate the fitness value of each particle by formula (9), update the personal best position P_i and obtain the global best position P_g across the population;

(5) If the stop criterion is met, go to step (7); or else go to step (6);

(6) Update the position vector of each particle according to (8);

(7) Output the P_g as a group of optimized parameters;

5 Experiments of QPSO-Trained RBF on System Identification

This section presents experiments on the application of QPSO-Trained RBF neural network in nonlinear system identification. First, we give a brief introduction of system identification model based on RBF NN. Then, the simulation results on a nonlinear system are presented.

Identifying a nonlinear system is the process of determining dynamic behavior of the system using observed input and output data. The identification model based on RBF NN is shown in Fig.1.

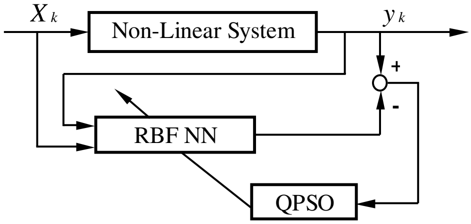


Fig. 1. System Identification Model based on RBF NN

Now consider a nonlinear system S to be identified, given that $D = \{(Y_k, X_k) | k = 1,2,\cdots,N\}$ is the observed data set, N is size of data set, and Y_k is the output corresponding to input X_k. For any input $X_i (1 \le i \le N)$, we can obtain the response $f_\theta(X_i)$ and compute the square error $(Y_i - f_p(X_i))^2/2$. Consequently, we can get the square error of the identifier on the whole data set.

$$MSE_{RBF}(D,\theta) = \frac{1}{2N}\sum_{i=1}^{N}(Y_i - f_\theta(X_i))^2 \tag{10}$$

where θ is the unknown parameter vector of the system. Thus the task of identification problem is to find out such a parameter vector $\overset{\wedge}{\theta} \in \Theta$ that

$$e_{RBF}(D,\overset{\wedge}{\theta}) = \min_{\theta\in\Theta} MSE_{RBF}(D,\theta) \tag{11}$$

where Θ is set of feasible parameter vector.

In our experiments on system identification, we use the following nonlinear system as a testing problem.

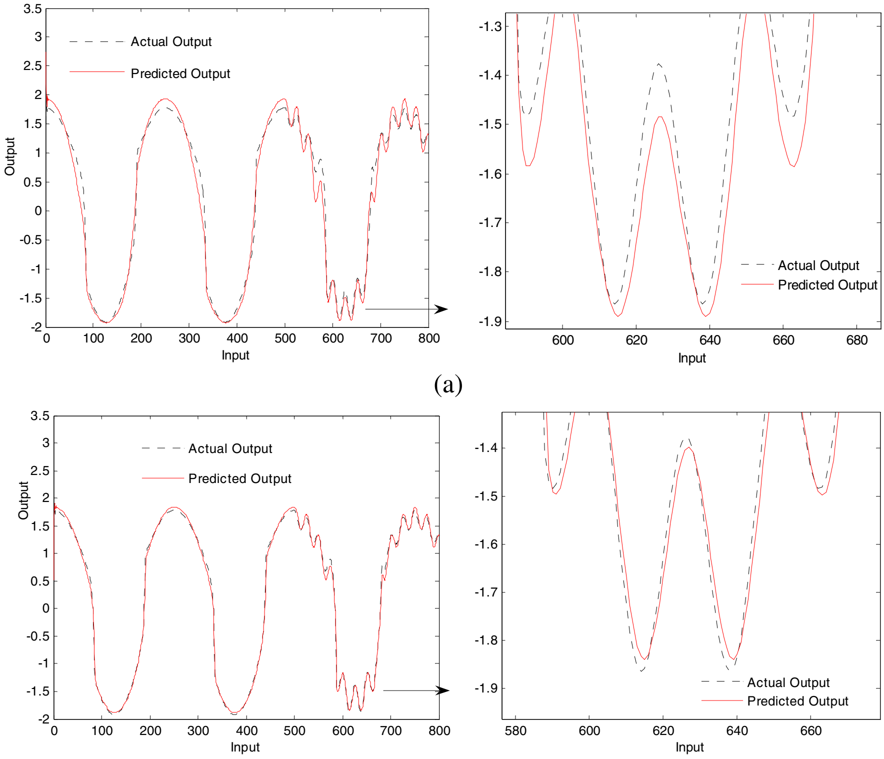
$$y_{k+1} = \frac{1.5y_t}{1+y_t^2} + 0.3\cos(y_t) + 1.2u_t \tag{12}$$

Given that u_t is a stochastic sequence uniformly distributed in the interval $[-2,2]$ and $x_t = [y_t, u_t]^T$ is the input of the identification model, the training set $D = \{(y,x)_k \mid k = 1,2...800\}$ is used to train the RBF NN identifier. First, we can determine the neuron number of hidden layer is 8 using RPCL algorithm. Then, we employ PSO and QPSO as training algorithm respectively and compare the performance between them. The experiment configuration is as follows. For PSO, the inertial weight w varies linearly from 0.9 to 0.4 over the running of the algorithm and the acceleration coefficients c_1 and c_2 are both set to 2; For QPSO, the CE coefficient varies linearly from 0.8 to 0.3 over the running. Both the training algorithms use 50 particles and execute for 200 iterations.

To test the results of identification, we use the following inputs

$$u_t = \begin{cases} \cos(2\pi t/250); & t \le 500 \\ 0.8\cos(2\pi t/250) + 0.2\cos(2\pi t/25) & t > 500 \end{cases} \tag{13}$$

The predicted outputs of the identification model trained by the two algorithms, along with the actual output of the system, are shown in Fig. 2. The convergence processes of the two algorithms are visualized in Fig. 3.



(a)

(b)

Fig. 2. The actual output and the predicted output of RBF NN identification model trained by (a) PSO and (b) QPSO

It can be seen from Fig.2 that, executing for same number of iteration, RBT NN identifier trained by QPSO can identify the nonlinear system with higher precision than that trained by PSO. Actually, QPSO can generate mean square error (MSE) value 3.1×10^{-3}, while PSO yield MSE value 1.0×10^{-2}. As of convergence speeds, we can see from Fig.3 that QPSO converges to the global optima more quickly than PSO at the early stage of the search process. During the middle stage, say about from the 15th iteration to the 30th iteration, PSO overruns QPSO in convergence. After the 30th iteration, QPSO overruns PSO again and could find better solution than PSO, while PSO may encounter premature convergence. Thus it can be concluded that QPSO outperforms PSO in global search ability.

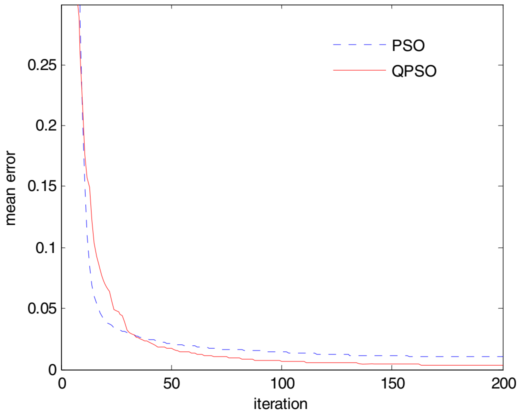


Fig. 3. Convergence process of PSO and QPSO over 200 iterations

In practice, the QPSO-Trained RBF neural network outperforms that trained by PSO not only in system identification, but also in function approximation problem and other application. In our preliminary experiments, we also use QPSO-Trained RBF neural network to approximate some well functions. The results, which are not presented here for space limitation, shows that QPSO-Trained RBF NN can approximated more rapidly and precisely than PSO-Trained RBF NN.

4 Conclusion

Radial Basis Function neural network have been widely used in many real world application. In order to use a RBF network, training algorithm is key to determine the network parameters. To overcome the shortcomings of existing training algorithms, in this paper, we employ the newly proposed QPSO to train RBF network and test the approach on system identification problem. The experiment result of QPSO-Trained RBF network on nonlinear system identification show that it can identify the system more quickly and precisely than that trained by PSO.

Although QPSO is also an evolutionary population-based search technique like PSO, it is stronger global search ability than PSO. Therefore, QPSO can find out the global optima of the optimization problem at more easily and more quickly. Our future work will focus on applying QPSO to training other neural network and use QPSO-Trained RBF in real world applications.

References

1. Van den Bergh, F.: An Analysis of Particle Swarm Optimizers. PhD Thesis. University of Pretoria, Nov. 2001.
2. Broomhead, D.S., Lowe, D.: Multivariable Functional Interpolation and Adaptive Networks. Complex Systems, vol. 2. (1998) 321-355
3. Casdagli, M.: Nonlinear Prediction of Chaotic Time Series. Physica D, vol. 35. (1989) 335-356
4. Chen, S., Cowan C.F.N., Grant, P.M.: Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. IEEE Transaction on Neural Networks, vol.2, no.2. (1991) 302-309
5. Clerc, M.: The Swarm and Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization. Proceedings of 1999 Congress on Evolutionary Computation. Piscataway, NJ (1999) 1951-1957
6. Clerc, M., Kennedy, J.: The Particle Swarm: Explosion, Stability, and Convergence in a Multi-dimensional Complex Space. IEEE Transactions on Evolutionary Computation, Vol. 6, No. 1. Piscataway, NJ (2002) 58-73
7. Kohonen, T.K.: Self-Organization and Associative Memory. Springer-Verlag, Berlin (1989)
8. Juang, C.F.: A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design. IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics, Vol. 34, No. 2. Piscataway, NJ (2004) 997-1006
9. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks, IV. Piscataway, NJ (1995) 1942-1948
10. Kennedy, J.: Stereotyping: Improving Particle Swarm Performance with cluster analysis. Proceedings of 2000 Congress on Evolutionary Computation. Piscataway, NJ (2000) 1507-1512
11. Sanner, R. M., Slotine, J.-J. E.: Gaussian Networks for Direct Adaptive Control. IEEE Transactions on Neural Networks, vol.3, no.6. Piscataway, NJ (1992) 837-863
12. Sun, J., Feng, B., Xu, W.B.: Particle Swarm Optimization with Particles Having Quantum Behavior. Proceedings 2004 Congress on Evolutionary Computation, Piscataway, NJ (2004) 325-331
13. Sun, J., Xu, W.B., Feng, B.: A Global Search Strategy of Quantum-behaved Particle Swarm Optimization. Proceedings of 2004 IEEE Conference on Cybernetics and Intelligent Systems. Singapore (2004) 111-116
14. Sun, J., Xu, W.B., Feng, B.: Adaptive Parameter Control for Quantum-behaved Particle Swarm Optimization on Individual Level. Proceedings of 2005 IEEE International Conference on Systems, Man and Cybernetics. Piscataway, NJ (2005) 3049-3054
15. Shi, Y., Eberhart R.C.: Empirical Study of Particle Swarm optimization. Proceedings of Congress on Evolutionary Computation. Piscataway, NJ (1999) 1945-1950
16. Shi, Y., Eberhart, R.C.: A Modified Particle Swarm. Proceedings of 1998 IEEE International Conference on Evolutionary Computation. Piscataway, NJ (1998) 1945-1950
17. Xu, L., Krzyzak, A., Oja, E.: Rival Penalized Competitive Learning for Clustering Analysis, RBF Net, and Curve Detection. IEEE Transactions on Neural Networks. Vol. 4, No. 4 (1993) 636-649
18. Moody, J.: Fast Learning in Networks of Locally-tuned Processing Units. Neural Computation, vol. 1, (1989) 281-294

Discrete Particle Swarm Optimization and EM Hybrid Approach for Naive Bayes Clustering*

Jing-Hua Guan, Da-You Liu, and Si-Pei Liu

Sch. of Computer S&T, Jilin Univ., Postfach 13 00 12, Changchun, China;
Key Lab. of Symbolic Computation & Knowledge Eng. of Ministry of Education, Jilin Univ.,
Postfach 13 00 12, Changchun, China
{gjh_jlu@hotmail.com, dyliu@mail. jlu.edu.cn,
lsp_jlu@email.jlu.edu.cn}

Abstract. This paper presents an improved Naive Bayes algorithm for clustering. Many researchers search for parameter values from incomplete data using EM (Expectation Maximization) algorithm. It is well-known that EM approach has a drawback – local optimal solution, so we propose a novel hybrid algorithm of the DPSO (Discrete Particle Swarm Optimization) and the EM approach to improve the global search performance. We then apply the approach to 4 real-world data sets from UCI repository and compare the performance of clustering by the new algorithm with by EM algorithm. In the comparison, the hybrid DPSO+EM algorithm exhibits more effectively and outperforms the EM approach.

Keywords: Naive Bayes; clustering; PSO; EM.

1 Introduction

Clustering [1] is the unsupervised classification of patterns into clusters, because unlike classification (known as supervised learning), no a priori labeling of some patterns is available to use in categorizing others and inferring the cluster structure of the whole data. The clustering problem has been addressed in many contexts and by researchers in many disciplines; this reflects its broad appeal and usefulness as one of the steps in data analysis. Many algorithms have been proposed, such as model-based algorithm, distance-based algorithm, density-based algorithm and deviation-based algorithm. In this paper, we concentrate on the research of the model-based algorithm. The most frequently used approaches include mixture density models (e.g., Gaussian mixture models [2]) and bayesian networks (e.g., AutoClass [3]).

In this paper, a hybrid approach is developed for the parameter estimation of Naive Bayes for clustering. We redefine position and velocity of PSO (Particle Swarm Optimization), and reinterpret the formula, which are described in details to adapt the parameter estimation problem from incomplete data. We have conducted a number of experiments and compare DPSO+EM with EM. Simply put, DPSO (Discrete

* Supported by the National High-Tech Research and Development Plan of China under Grant(863) No.2001AA115160.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1164 – 1173, 2006.

PSO)would take less time for finding the optimal solution. The empirical results illustrate that DPSO+EM can generate more effective clustering results than the EM algorithm.

This paper is organized as follows. We present the backgrounds of Naive Bayes Clustering, the Expectation maximum algorithm and the Particle Swarm Optimization method in Section 2. We derive a novel hybrid DPSO algorithm for clustering in detail in section 3 and describe a comparison among this new algorithm and EM algorithm with an analytical study in Section 4. We conclude with a discussion of future directions in section 5.

2 Backgrounds

2.1 Naive Bayes Clustering

NB (Naive Bayes) can be viewed as a special example of Bayesian Network. It assumes that all variables are conditionally independent given the class variable. Independence means probabilistic independence, that is, X_1 is independent of X_2 given C where $P(X_1| X_2,C) = P(X_1|C)$ for all possible values of X_1, X_2 and C, whenever $P(C)$>0. NB is applied in many domains, such as classification [4], clustering etc on account of its efficiency.

AutoClass is assumed that, in addition to the observed or predictive attributes, there is a hidden variable. This unobserved variable reflects the cluster membership for every case in the data set. Therefore, the data-clustering problem is also an example of supervised learning from incomplete data due to the existence of such a hidden variable [5]. Their approach for learning has been called RBMNs (Recursive Bayesian Multinets). These two algorithms use the Bayesian approach, starting from a random initialization of the parameters, incrementally adjusts them in an attempt to find their maximum likelihood estimates. As Friedman points in [6], the computation of the MAP (Maximum a Posterior) parameters can be done efficiently using the EM algorithm, gradient ascent, Gibbs Sampling or extensions of these methods, such as BC+EM algorithm provided in [7] [8].

2.2 EM Algorithm

The well-known EM [9][10] algorithm is an iterative method to compute maximum a posteriori and maximum likelihood parameters from incomplete data. The EM algorithm finds a local maximum for the parameters. The traditional EM algorithm is a process with two steps:

$$\text{Estep: } Q(\Theta \big| \Theta^{(t)}) \triangleq E\{\mathcal{L}_c(\Theta;\mathcal{Z})\big|\mathcal{X};\Theta^{(t)})\} \tag{1}$$

$$\text{Mstep: } \Theta^{(t+1)} = \arg\max_{\Theta} Q(\Theta,\Theta^{(t)}) \tag{2}$$

where $\mathcal{Z}$ is a set, which consists of observed data $\mathcal{X}$ and unobserved data $\mathcal{Y}$. $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$ and $\mathcal{X}$ are called complete data and incomplete data, respectively. Assume that the joint probability density of $\mathcal{Z}$ is parametrically given as $p(\mathcal{X}, \mathcal{Y}; \Theta)$, where Θ denotes parameters of the density to be estimated. The maximum likelihood estimate of Θ is a value of Θ that maximizes the incomplete data log-likelihood function:

$$\mathcal{L}(\Theta;\mathcal{X}) \triangleq \log p(\mathcal{X};\Theta) = \log \int p(\mathcal{X},\mathcal{Y};\Theta)d\mathcal{Y} \tag{3}$$

The characteristic of the EM algorithm is to maximize the incomplete data log-likelihood function by iteratively maximizing the expectation of the complete data log-likelihood function:

$$\mathcal{L}_c(\Theta;\mathcal{Z}) \triangleq \log p(\mathcal{X},\mathcal{Y};\Theta) \tag{4}$$

Suppose that $\Theta^{(t)}$ denotes the estimate of Θ obtained after the tth iteration of the algorithm. Then, at the $t+1$th iteration, the Estep computes the expected complete data log-likelihood function denoted by $Q(\Theta|\Theta^{(t)})$ and defined by equation (1). And the M-step finds the Θ maximizing $Q(\Theta|\Theta^{(t)})$. The convergence of the EM steps is theoretically guaranteed [9]. By these two steps we can get the MLE (maximum likelihood estimation) of Θ. It is well-known that the EM algorithm is sensitive to the initialization and easy to get into local optima.

2.3 PSO Algorithm

PSO is basically developed through simulation of bird flocking in two-dimension space [11]. The basic principles in “classical” PSO are very simple. A set of moving particles (the swarm) is initially “thrown” inside the search space. Each particle has the following features:

- It has a position and a velocity
- It knows its position, and the objective function value for this position
- It knows its neighbors, best previous position and objective function value (variant: current position and objective function value)
- It remembers its best previous position

This compromise is formalized by the following equations:

$$v^{t+1} = \omega v^{t} + c_1 r_1 (p^{t}_{pBest} - x^{t}) + c_2 r_2 (p^{t}_{gBest} - x^{t}) \tag{5}$$

$$x^{t+1} = x^{t} + v^{t+1} \tag{6}$$

where
v^t: velocity of particle at iteration t
x^t: position of particle at iteration t
p^t_{pBest}: best previous position of particle at iteration t
p^t_{gBest}: best neighbor’s position of particle at iteration t
ω: inertia weight
r_1, r_2: random coefficient in [0, 1]
c_1, c_2: positive constant

In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles, namely only the particle with the best fitness function value can transfer information to the others. So these particles could quickly converge at the optimal solution. In addition, it performs a globalized search for solution whereas most other optimization algorithms perform a localized search. It can be used to solve a wide array of different optimization problems. Some example application include neural network training [12][13] and function minimization [14][15].

3 DPSO+EM Parameter Estimation

In past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods on a continuous definition domain. In addition, some binary versions have already been used. Naturally, we try to define a frame for a discrete PSO to optimize Naive Bayes parameter estimation for clustering. By means of the character of this problem, we need redefine velocity and position, and reinterpret the formulas (5) and (6).

3.1 DPSO Algorithm

DPSO approach, motivated by swarm behavior, makes use of velocity and position to obtain the globally optimal partition of the data. Candidate solutions to the clustering problem are encoded as position of particles. The summary of DPSO approach applied to clustering is as follows:

1. Choose a random population of solutions. Each solution here corresponds to a valid k-partition of the data. Associate a fitness value with each solution. In terms of fitness function, calculate fitness of each particle.
2. According to fitness of particle, select the best neighbor particle with the smallest fitness. Use the DPSO operators to generate the next population of solutions. Evaluate the fitness values of these solutions.
3. Repeat step 2 until some termination condition is satisfied.

3.1.1 Position and Velocity Redefinition

In order to apply the DPSO to clustering problem, we redefine the position and velocity which are used in PSO as follows.

A position is defined by $\boldsymbol{x} = (x_1, \ldots, x_i, \ldots, x_N)$($N$ is the number of cases), $x_i \in \{1, \ldots, K\}$(K is the number of clusters), which means the ith case in data set is assigned to the x_ith cluster.

A velocity is then defined by $\boldsymbol{v} = (v_1, \ldots,. v_N)$, $v_i = (v_{i1}, \ldots, v_{ik}, \ldots, v_{i\mathrm{K}})$, $v_{ik} \in (0,1)$, which means the probability that the ith case is assigned to the x_ith cluster.

3.1.2 Formulas Reinterpretation

1. position plus velocity

Let $\boldsymbol{p}$ be a position and $\boldsymbol{v}$ be a velocity. The position $\boldsymbol{p}' = \boldsymbol{p} + \boldsymbol{v}$ is found by applying the first transposition of $\boldsymbol{v}$ to $\boldsymbol{p}$, then the second one to the result etc.

Let $k' = \arg\max_{k}(v_{ik})$ and r be a random coefficient in [0, 1]

If $v_{ik} > r$ then $p_i' = k'$

else $p_i' = p_i$

2. position minus position

Let $\boldsymbol{p}^1$ and $\boldsymbol{p}^2$ be two positions. The difference $\boldsymbol{p}^2$-$\boldsymbol{p}^1$ is defined as the velocity v, found by a given algorithm, so that applying v to p_1 obtains p_2.

$\boldsymbol{v} = \boldsymbol{p}^2\text{-}\boldsymbol{p}^1 = (p^2{}_1, \ldots, p^2{}_N) - (p^1{}_1, \ldots., p^1{}_N) = ((v_{11}, \ldots, v_{1k}, \ldots, v_{1K,}), \ldots, (v_{i1}, \ldots, v_{ik}, \ldots, v_{iK}), \ldots, (v_{N1}, \ldots, v_{Nk}, \ldots, v_{NK}))$

If $p^2{}_i = p^1{}_i$ then $v_{ik} = 2\times1/N$, $v_{ij} = 0$ (j =1, …, k-1,k+1, …, K)

else when $p^2{}_i = k$ then $v_{ik} = 1/N$, $v_{ij} = 0.5\times1/N$ (j =1, …, k-1,k+1, …, K)

3. ω×velocity plus r_1×velocity1 plus r_2×velocity2

Let $\boldsymbol{v}^1$ and $\boldsymbol{v}^2$ be two velocities and r_1 and r_2 be two real. We define

$\boldsymbol{v}' = \omega\times\boldsymbol{v} + r_1\times\boldsymbol{v}^1 + r_2\times\boldsymbol{v}^2 = (\omega\times v_1 + r_1\times v^1{}_1 + r_2\times v^2{}_1, \ldots, \omega\times v_i + r_1\times v^1{}_i + r_2\times v^2{}_i, \ldots, \omega\times v_N + r_1\times v^1{}_N + r_2\times v^2{}_N)$

$v_i' = \omega\times v_i + r_1\times v^1{}_i + r_2\times v^2{}_i = (\omega\times v_{i1} + r_1\times v^1{}_{i1} + r_2\times v^2{}_{i1}, \ldots, \omega\times v_{ik} + r_1\times v^1{}_{ik} + r_2\times v^2{}_{ik,} \ldots, \omega\times v_{iK} + r_1\times v^1{}_{iK} + r_2\times v^2{}_{iK})$

Finally, in order to make sure that the velocity (probability) is normalized, we calculate velocity using $v_{ik}' = v_{ik}' \big/ \sum_k v_{ik}'$.

3.1.3 Fitness Function

The MDL (Minimal description length) principle casts learning in terms of data compression. Roughly speaking, the goal of the learner is to find a model that facilitates the shortest description of the original data. The length of this description takes into account the description of the model itself and the description of the data using the model. In this paper, the model is a NB network. Because of the structure of model is known, the main problem is learning parameters of NB network according to the MDL principle.

Let B = <S, Θ> be a NB network, S and Θ are the structure and parameters of NB model, respectively. Let D be an incomplete data set. The MDL score function of a network B given a data set D, written $MDL(B|D)$, as stated in [3] is given by:

$$MDL(B|D) = \frac{\log N}{2}|B| - N\sum_{i=1}^{n}\sum_{\substack{x_i\in Val(X_i)\\ c_j\in Val(C)}} \hat{P}_D(x_i, c_j)\log \hat{P}_D(x_i|c_j) \qquad (6)$$

Where |B| denotes the number of parameters of NB model, and N , n denote the number of the cases in data set and the number of variable, respectively. The first term represents the length of describing the NB model B, in that, it counts the bits needed to encode the specific NB model B, where $(\log N)/2$ bits are used for each parameter in Θ. The second term is the negation of the log likelihood of B given D, which measures how many bits are needed to describe D based on the probability distribution $P_{B.}$ Let $\hat{P}_D(\cdot)$ be the empirical distribution defined by frequencies of events in D.

The MDL principle is widely applied to model selection problems. In this paper, we refer to MDL score as fitness function of DPSO algorithm, so the smaller fitness function is, the better corresponding to particle is.

3.1.4 Ascent Criterion

In order to make information share within these particles in a swarm more effective, we propose to unify the positions and velocities of each particle using ascent criterion after they are updated by the DPSO algorithm.

Let $D = \{d_1, \ldots, d_N\}$ represent a collection of N objects. Let partition = $(C_1, \ldots, C_K)$ denote a partition of D into K nonempty clusters. Each set C_i consists of $n_i \geq 1$ elements of D, with $\sum_{i=1}^{K} n_i = N$. Without loss of generality it will be assumed that $C_1, \ldots, C_K$ are "sorted in ascending order" which means that:

$$\arg\min_i (d_i \in C_1) < \arg\min_i (d_i \in C_2) < \ldots < \arg\min_i (d_i \in C_K) .$$

3.2 DPSO+EM Algorithm

The well-known EM algorithm is an iterative approach to compute maximum a posteriori and maximum likelihood parameters from incomplete data. The EM algorithm finds a local maximum for the parameters because it is sensitive to initialization.

The DPSO method, described above, is an evolutional method to estimate conditional probabilities from incomplete data sets. It is obvious that the DPSO method has strong global search ability and parallel performance, but the convergence rate of the DPSO algorithm is painfully slow in terms of experiments. We present an alternative approach to solve NB parameter estimation problem. In the remaining part of this paper we refer to this method as DPSO+EM as it alternates between the DPSO method and the EM algorithm. The DPSO+EM method overcomes the disadvantages of these two approaches by performing a local optimization using the EM method at each particle

The general pseudo code of DPSO+EM algorithm for parameter estimation from incomplete data set can be described as appendix.

4 Experimental Results

In our experiments, we compared the DPSO+EM clustering algorithm with AutoClass over 4 data sets. These 4 real data sets we used are heart_disease (270 cases, 2 classes, 13 attributes), iris (150 cases, 3 classes, 4 attributes), nursery (resample 389 cases from initial data set, 5 classes, 8 attributes) and zoo (101 cases, 7classes, 16 attributes) from the UCI repository.

In all experiments we assume that the class label of each case is not given and all the variables are finite discrete variables. Before performing the clustering process, we need pretreatment to data sets. First, we discarded all the entries corresponding to the class variable. Second, the observed data were discretized using fixed-sized bins. Notice should be taken that in all experiments we assume that the number of clusters is unknown, thus, we perform a search step to identify the most probable number of clusters in these data sets. We use 8 as maximum of the number of clusters and calculate the MDL score for different number. The smaller MDL score is, the better the number of partition is. This algorithm favors smaller number of clusters when the MDL score is similar (Occam's razor). In experiments, we choose 20 particles for the DPSO algorithm. In the DPSO algorithm, the inertia weight ω is initially set as 1 and the acceleration coefficient constants c_1 and c_2 are set as 2. These values are chosen based on the experimental results.

We use two performance criteria to compare the EM with the DPSO+EM algorithms. The log likelihood (LL) of the learnt NB model, log $p(D|\Theta)$, is used in our comparison. The higher LL is, the closer NB model is to modeling the probability distribution in the data set D. Notice should be taken in experiments, we assume that the real number of each dataset is the optimal number of each dataset, and then estimate parameters of NB model using EM and DPSO+EM algorithms, respectively. Because these two algorithms use the same optimal number of clusters, we don't need to use MDL to compare performance between these algorithms, LL can be viewed as a performance comparison criterion according to the description of MDL in section 3.

In addition to this, we consider the optimal number of clusters and the corresponding MDL score as valuable comparison information. In section 3, we describe the meaning and performance of MDL principle in parameter estimation.

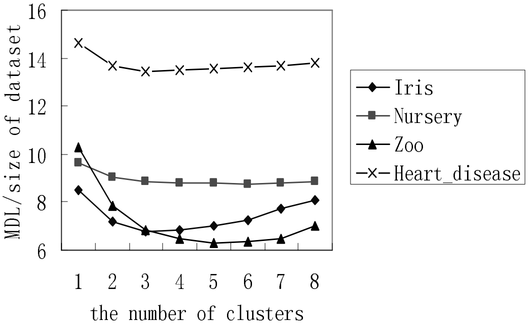


Fig. 1. The mean of MDL score of learnt NB model for 4 data sets

Figure 1 shows the curves of MDL score changing along with different number of clusters over 4 data sets. These curves are single peak, so we can confirm the number of clusters according to MDL score. We also find that the MDL score over iris data set is minimum using the DPSO+EM algorithm when the number of clusters is 3. It is interesting that the real number is also 3. This illustrates that this algorithm is effective to find the optimal number of clusters and the number has significance for iris data set.

Table 1 compares the performance of the algorithm for learning NB model for clustering when using the EM algorithm as parameter search step and when using the DPSO+EM method. It shows that, in most data sets, the DPSO+EM algorithm can effectively confirm the number of clusters of data set rather than EM algorithm, and corresponding MDL score is smaller than the EM algorithm's. The primary reason is that the EM algorithm can get good result on single pick function, however when the distribution is multi-peak function, the solution of optimization is not good. As we can see in the 4 data sets, the use of the DPSO+EM method outperforms the use of the EM algorithm in terms of log marginal likelihood.

Table 1. Clustering performance comparison bewteen EM and DPSO+EM

Data Set	number of clusters			MDL		Log likelihood	
	EM	DPSO+EM	Real	EM	DPSO+EM	EM	DPSO+EM
heart_disease	4	3	2	3709.7	3638.8	-3693.6	-3436.4
iris	5	3	3	1052.5	1005.8	-844.5	-739.5
nursery	4	4	5	3534.3	3402.3	-3170.4	-3045.6
zoo	4	3	7	626.6	644.9	-405.0	-352.1

We can conclude that the DPSO+EM algorithm is not sensitive to the initial solution, and can get better solution than the EM algorithm.

Due to the above experimental results, the DPSO+EM method exhibits a more effective, efficient and robust behavior than the EM algorithm.

5 Conclusion

In this paper, we have described how to apply DPSO method to parameter estimation for NB clustering effectively. We found that the DPSO algorithm converges much slowly than the EM algorithm according to experimental results, but it can get better global optimal solution. At the same time, the EM method is sensitive to initial solution and easy to get local optima. Because DPSO and EM algorithms have their own drawbacks respectively, in order to improve the efficiency of DPSO, local search algorithm-EM is introduced into the traditional DPSO method. The EM algorithm makes every particle can find the local optimal solution in current space. This local search process improves the performance of swarm.

This hybrid algorithm is tested on 4 dataset to show that it can discover good clustering results. We have evaluated and compared the EM and the DPSO+EM algorithms for data clustering problems. Our experimental comparison between both algorithms has suggested the substantial gain in effectiveness and efficiency of the DPSO+EM algorithm over the EM algorithm.

We are currently extending DPSO+EM to recognize outliers from data set; we hope this will improve the performance of outlier detection.

References

1. Kotsiantis, P. Pintelas, Recent Advances in Clustering: A Brief Survey, *WSEAS Transactions on Information Science and Applications*, Vol 1, No 1 (73–81), 2004.
2. Banfield J., Raftery A., Model-based Gaussian and non-Gaussian Clustering. *Biometrics*, 49, 803-821, 1993.
3. Cheeseman P., Stutz J.,Bayesian classification (Auto-Class): Theory and results. *Advances in Knowledge Discovery and Data Mining*. AAAI Press, Menlo Park, CA, pp. 153–180, 1995.
4. Friedman N., Geiger D., Goldszmidt M., Bayesian Network Classifiers, *Machine Learning* 29:131—163, 1997.
5. Pena J., Lozano J., Larranaga P., Learning Recursive Bayesian Multinets for Data Clustering by Means of Constructive Induction, *Machine Learning*, 47, 63–89, 2002.
6. Friedman N., The Bayesian Structural EM algorithm. *Proc. 14th Conf. on Uncertainty in Artificial Intelligence.* Morgan Kaufmann, San Francisco, CA, pp. 129–138, 1998.
7. Pena J.M., Lozano J.A., Larranaga P., Learning Bayesian networks for clustering by means of constructive induction, *Pattern Recognition Letters 20*, 1219-1230, 1999.
8. Pena J.M., Lozano J.A., Larranaga P., An improved Bayesian structural EM algorithm for learning Bayesian networks for clustering, *Pattern Recognition Letters 21*, 779–786, 2000.
9. Dempster A.P., Laird N. M., Rubin D.B., Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B*, 39, 1-38, 1977.
10. McLachlan G. J., Krishnan T., The EM Algorithm and Extensions, *Wiley*, New York, 1997.
11. Kennedy J., Eberhart R., Particle Swarm Optimization, *Proc. of IEEE International Conference on Neural Networks*, Vol. IV, pp.1942-1948, Perth, Australia, 1995.
12. A. P. Engelbrecht and A. Ismail, “Training product unit neural networks”, *Stability and Control: Theory and Application*s, Vol. 2, No. 1-2, 59-74, 1999.
13. F. van den Berg, “Particle Swarm Weight Initialization in Multi-layer Perceptron Artificial Neural Networks”, In *Development and Practice of Artificial Intelligence Technique*s, 41-45, 1999.
14. Y. Shi and R.C. Eberhart, “Empirical Study of Particle Swarm Optimization”, Proc. Of *the Congress on Evolutionary Computation*, 1945-1949, 1999.
15. Y. Shi and R.C. Eberhart,, “A Modified Particle Swarm Optimizer”, *IEEE International Conference of Evolutionary Computation*, May 1998.

Appendix

```
k=2
Repeat:
  Create and randomly initialize n N-dimension particles $P_1$-$P_n$
  Repeat:
    t = 1
    For i = 1 to n
      Optimize $P_i$'s position using EM method until convergence or EM iteration's number > 5
      Update position and velocity of particle $P_i$ in terms of ascent criterion
      Calculate $P_i$'s MDL Score as its fitness value
    Next
    Select the particle with the best fitness as gBest[1]
    For i=1 to n
      Update $P_i$'s velocity and position in terms of equations (5),(6)and gBest
      Update position and velocity of $P_i$ according to ascent criterion
      Calculate current $P_i$'s MDL Score
      If MDL($P_i$)<MDL(pBest[2])
        Set current $P_i$ as $P_i$'s new pBest
    Next
    t = t+1
  Until convergence criterion is satisfied or maximum iteration number is attained
  k = k + 1
Until maximum of the number of clusters is attained or convergence criterion is satisfied.
End.
```

[1] best neighbor particle at iteration t.

[2] best previous particle at iteration t.

Extended Particle Swarm Optimiser with Adaptive Acceleration Coefficients and Its Application in Nonlinear Blind Source Separation*

Ying Gao, Zhaohui Li, Hui Zheng, and Huailiang Liu

Department of Computer Science and Technology Guangzhou University,Guangzhou, 510006, P.R. of China
falcongao@21cn.com

Abstract. First, based on the particle swarm optimization, an extended particle swarm optimizer with acceleration coefficients (EPSO_AAC) is presented. The personal best particle is replaced by the average of personal best particles in swarm at generation, and time-varying acceleration coefficients are applied by establishing a nonlinear functional relationship between acceleration coefficients and the different of the average fitness of all particles and the fitness of the global best particle. The proposed algorithm uses more particles' information, and adjusts adaptively "cognition" component and "social" component by time-varying acceleration coefficients, thus improves convergence performance. Then, the proposed algorithm is applied to nonlinear blind source separation. The demixing system of the nonlinear mixtures is modeled using a multi-input multi-output B-spline neural network whose weights are optimized under the criterion of independence of its outputs by EPSO_AAC. The experiment results demonstrate that the proposed algorithms are effective, and have good convergence performance.

1 Introduction

The particle swarm optimization (PSO), first introduced by Kennedy and Eberhart[1,2] in 1995, is a stochastic optimization technique. They have been used to solve a range of optimization problems, including neural network training and function minimization. The PSO is a population based optimization algorithm. Similar to other population-based optimization methods such as genetic algorithms, the PSO starts with the random initialization of a population of individuals (particles) in the search space. It works on the social behavior of particles in the swarm. Therefore, it finds the global best solution by simply adjusting the trajectory of each individual toward its own best

* The work is supported by the Post Doctor Science Foundation of P.R.C. (2003034062), the Natural Science Foundation of Guangdong Province, P.R.C. (04300015) , the Program for the Development of Science & Technology of Guangzhou, P.R.C.(2004J1-C0323) and the Program for the Development of Science & Technology of Guangzhou Colleges and Universities, P.R.C.(2055).

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1174 – 1182, 2006.

location and toward the best particle of the entire swarm at generation. The PSO is becoming very popular due to its simplicity of implementation and ability to quickly converge to a reasonably good solution. Since the introduction of the PSO in 1995,there has been a considerable amount of work done in developing the original version of PSO. Recently, several investigations have been undertaken to improve the performance of PSO, such as He[3], Ratnaweera[4], Monson[5], van[6], Rodriguez[7], Kennedy[8], Katare[9], Fan[10] and Sun[11].

In this paper, first, we present an extended particle swarm optimizer with adaptive acceleration coefficients (EPSO_AAC) to improve the performance of standard PSO. By replacing personal best particle with the average of personal best particles in swarm, more information can be transferred among particles of swarm. Then, adaptive acceleration coefficients are introduced by establishing a nonlinear functional relationship between acceleration coefficients and the different of the average fitness of all particles and the fitness of the global best particle. The proposed algorithm uses more particles' information, and adjusts adaptively cognition component and social component, thus improves convergence performance. Then, the proposed algorithm is applied to nonlinear blind source separation. A multi-input multi-output B-spline neural network is used to model the demixing system of the nonlinear mixtures, and its weights are optimized under the criterion of independence of its outputs by EPSO_AAC. Application of the EPSO_AAC on several benchmark optimization problems shows a marked improvement in performance over original particle swarm optimization, and is effective to be applied to nonlinear blind source separation.

2 Particle Swarm Optimization

In the particle swarm optimization, the trajectory of each particle in search space is adjusted by dynamically altering the velocity of each particle, according to its own flying experience and the flying experience of the other particles in the search space. The position vector and the velocity vector of ith particle in m-dimensional search space can be represented as $\mathbf{x}_i(i=1,2,\cdots,N)$ and $\mathbf{v}_i(i=1,2,\cdots,N)$ respectively, N is the number of particle. In each iteration of PSO, the swarm is updated by the following equations:

$$\mathbf{v}_i(t+1)=w\mathbf{v}_i(t)+c_1r_1(\mathbf{p}_i(t)-\mathbf{x}_i(t))+c_2r_2(\mathbf{p}_g(t)-\mathbf{x}_i(t)) \tag{1}$$

$$\mathbf{x}_i(t+1)=\mathbf{x}_i(t)+\mathbf{v}_i(t+1) \tag{2}$$

Where $\mathbf{p}_i(t)(i=1,2,\cdots,N)$ and $\mathbf{p}_g(t)$ are given by the following equations, respectively:

$$\mathbf{p}_i(t+1)=\begin{cases}\mathbf{p}_i(t), & f(\mathbf{x}_i(t+1))<f(\mathbf{p}_i(t))\\ \mathbf{x}_i(t+1), & f(\mathbf{x}_i(t+1))\geq f(\mathbf{p}_i(t))\end{cases} \tag{3}$$

$$\mathbf{p}_g(t)\in\left\{\mathbf{p}_1(t),\mathbf{p}_2(t),\cdots,\mathbf{p}_N(t)\middle|f(\mathbf{p}_g(t))=\max\{f(\mathbf{p}_1(t)),f(\mathbf{p}_2(t)),\cdots,f(\mathbf{p}_N(t))\}\right\} \tag{4}$$

w is called an inertia weight. c_1 and c_2 are acceleration coefficients which control how far a particle will move in a single iteration. r_1 and r_2 are elements from two uniform random sequences in the range $[0,1]$. $f(\mathbf{x})$ is the maximum objective function.

The second part of (1), known as the "cognitive" component, represents the personal thinking of each particle. The cognitive component encourages the particles to move toward their own best positions found so far. The third part is known as the "social" component, which represents the collaborative effect of the particles, in finding the global optimal solution. The social component always pulls the particle toward the global best particle found so far.

3 Extended PSO with Adaptive Acceleration Coefficients (EPSO_AAC)

In the traditional particle swarm optimization, the sharing of information among conspecifics is achieved by employing the publicly available information $\mathbf{p}_g(t)$. There is no information sharing among individuals except that $\mathbf{p}_g(t)$ broadcasts the information to the other particles. Therefore, the swarm may lose diversity and is more likely to confine the search around local minima if committed too early in the search to the global best found so far. For using more particles' information, (1) is modified as:

$$\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + c_1 r_1(\mathbf{p}_a(t) - \mathbf{x}_i(t)) + c_2 r_2(\mathbf{p}_g(t) - \mathbf{x}_i(t)) \tag{5}$$

Where $\mathbf{p}_a(t) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_i(t)$.

On the other hand, the search toward the optimum solution is guided by the two stochastic acceleration coefficients (the cognitive component and the social component). Therefore, proper control of these two components is very important to find the optimum solution accurately and efficiently. Generally, during the early stages of optimization, it is desirable to encourage the particles to wander through the entire search space, without clustering around local optima, during the latter stages, it is important to enhance convergence toward the global optima, to find the optimum solution efficiently. Therefore, we reduce the acceleration coefficient c_1 and increase the acceleration coefficient c_2 by establishing a nonlinear functional relationship between acceleration coefficients and different of average fitness and best fitness. The mathematical representation of this concept is given as follow:

$$c_1(t) = 1 - \exp\left(-\left(F_a(t) - F_g(t)\right)^2\right) \tag{6}$$

$$c_2(t) = 1 - c_1(t) \tag{7}$$

Where $F_a(t) = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i(t))$, $F_g(t) = f(\mathbf{p}_g(t))$.

4 Nonlinear Blind Source Separation Based on EPSO_AAC

A generic nonlinear mixture model for blind source separation[12-13] can be described as

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{As}(t)) \tag{8}$$

Where A is a mixing matrix, $\mathbf{s}(t)=[s_1(t) \;\; s_2(t) \;\; \cdots \;\; s_n(t)]^{\mathrm{T}}$ called the independent source vector, $\mathbf{x}(t)=[x_1(t) \;\; x_2(t) \;\; \cdots \;\; x_n(t)]^{\mathrm{T}}$ called vector of observed random variables. $\mathbf{f}=[f_1(\cdot), f_2(\cdot), \cdots, f_n(\cdot)]^{\mathrm{T}}$, $x_i(t)=f_i(s_1(t), s_2(t), \cdots, s_n(t)), \quad i=1,2,\cdots,n$.

The output of the nonlinear separating system can been written as

$$\mathbf{y}(t) = \mathbf{Wg}(\mathbf{x}(t)) \tag{9}$$

Substituting (8) into (9), We can obtain

$$\mathbf{y}(t) = \mathbf{Wg}(\mathbf{f}(\mathbf{As}(t))) \tag{10}$$

If $\mathbf{g}(\cdot)=\mathbf{f}^{-1}(\cdot)$ and $\mathbf{WA} = \mathbf{PD}$, then this means that the components of the outputs $\mathbf{y}$ are independent. Where $\mathbf{P}$ is a permutation matrix and $\mathbf{D}$ is a nonsingular and diagonal matrix.

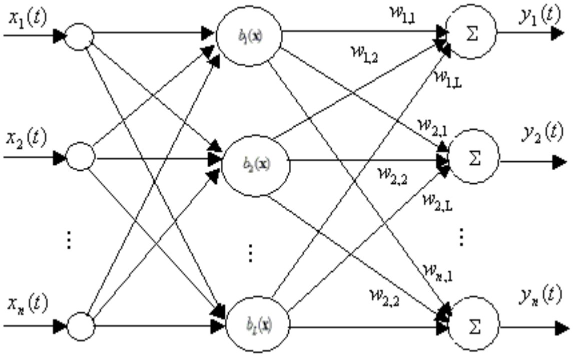


Fig. 1. B-spline neural network

For the nonlinear mixing transform function $\mathbf{f}$, we assume it has the inverse function $\mathbf{f}^{-1}$. $\mathbf{y}(t) = \mathbf{Wg}(\mathbf{x}(t))$ is a multi-input multi-output system , a B-spline neural network[14-15] of Fig.1 is used to approximate the unknown multi-input multi-output system.

Where $b_i(\mathbf{x}), i = 1, \cdots, \mathrm{L}$ is B-spline basic function. The outputs of the B-spline neural network showed in Fig.1 can been written as

$$y_i(t) = \sum_{j=1}^{L} w_{i,j} b_j(\mathbf{x}(t)) \qquad i = 1,2,\cdots,n \tag{11}$$

Where $w_{i,j}(i = 1,\cdots,n,\ j = 1,\cdots,\mathrm{L})$ are B-spline neural network weights, L is the number of B-spline neural network basic functions.

It is possible to recover the source from the nonlinear mixture (8) using only the source statistical independence assumption[12]. In order to separate the independent sources from their nonlinear mixtures, we expect the outputs of the separation system to be mutually statistically independent. For this purpose, we utilize the following criterion for nonlinear blind sources separation[16].

$$J(\mathbf{W}) = \sum_{i=1}^{n} \sum_{j \neq i}^{n} \left[\mathrm{E}\left(h_1(y_i)h_2(y_j)\right) - \mathrm{E}\left(h_1(y_i)\right)\mathrm{E}\left(h_2(y_j)\right)\right]^2 \tag{12}$$

Where $\mathbf{W}$ is weight matrix of the B-spline neural network, $h_1(\cdot)$ and $h_2(\cdot)$ are nonlinear function.

The minimization of the criterion in (12) can give the correct separation results for nonlinear mixtures. Here, we apply EPSO_AAC to fulfill the search of the optimal weighs of the separation system based on the cost functions specified in (12).

The PSO_AAC-based nonlinear BSS algorithm can be implemented as the following iterative procedure:

1) Initial population $\{\mathbf{W}_i\}_{i=1}^{N}$ and $\{\mathbf{v}_i\}_{i=1}^{N}$.
2) The fitness for each particle $\{\mathbf{W}_i\}_{i=1}^{N}$ is evaluated using (12),
3) The best previous position of the ith particle is recorded and represented as $\mathbf{p}_i(i = 1,2,\cdots,N)$, and the index of the best particle among all the particles in the population is represented as $\mathbf{p}_g$.
4) Apply EPSO_ACC to change the velocity and position vector for each particle.
5) Go to step 2), and repeat until convergence.
6) Output the particle $\mathbf{W}$ with the best fitness value and compute the separated signals.

5 Results from Simulations

In our experimental studies, a set of 5 benchmark functions was employed to evaluate the EPSO_AAC algorithm in comparison with PSO. The dimension of each function M=20. The population size of all algorithms used in our experiments was set at 100. The acceleration coefficients c_1 and c_2 for PSO were both 1.0. The inertia weight w for PSO and EPSO_AAC was 0.8. All experiments were repeated for 50 runs. A fixed number of maximum generations 500 was applied to all algorithms.

The experimental results for PSO and EPSO_AAC on each test function are listed in Table1 and showed in Fig.2-Fig.6. From Table1, EPSO_AAC outperformed the PSO algorithm significantly for 5 benchmark functions. From Fig.2-Fig.6, it can be seen that EPSO_AAC has a faster convergence rate than the PSO.

1) $f_1(\mathbf{x})=\sum_{i=1}^{M} x_i^2$, $\min f(\mathbf{x}^*) = f(0,0,\cdots,0) = 0$

2) $f_2(\mathbf{x})=\sum_{i=1}^{M}[x_i^2 - 10\cos(2\pi x_i)+10]$, $|x_i| \le 5.12$, $\min f_2(\mathbf{x}^*) = f(0,0,\cdots,0) = 0$

3) $f_3(\mathbf{x})=\frac{1}{4000}\sum_{i=1}^{M} x_i^2 - \prod_{i=1}^{M}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$, $|x_i| \le 600$, $\min f_3(\mathbf{x}^*) = f(0,0,\cdots,0) = 0$

4) $f_4(\mathbf{x}) = \sum_{i=1}^{M}\left(\sum_{j=1}^{i} x_j\right)^2$, $|x_i| \le 100$, $\min f_4(\mathbf{x}^*) = f(0,0,\cdots,0) = 0$

5) $f_5(\mathbf{x}) = \sum_{i=1}^{M}|x_i| + \prod_{i=1}^{M}|x_i|$, $|x_i| \le 10$, $\min f_5(\mathbf{x}^*) = f(0,0,\cdots,0) = 0$

Table 1. Comparison between PSO and EPSO_AAC

function	Averge Optimum Solution	
	PSO	EPSO_AAC
f_1	2.5581E-4	7.7941E-6
f_2	17.9678	3.5243
f_3	2.2645E-4	1.9146E-4
f_4	0.0450	0.0120
f_5	0.4980	0.0990

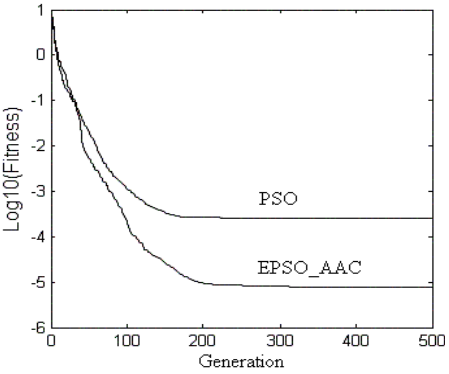


Fig. 2. The convergence curves of f_1

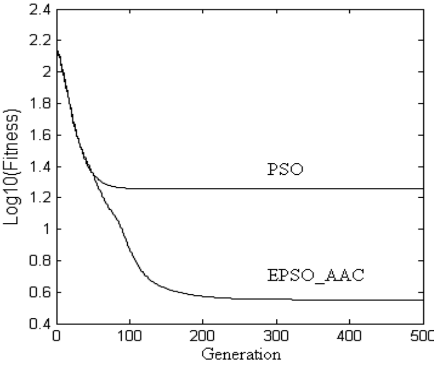


Fig. 3. The convergence curves of f_2

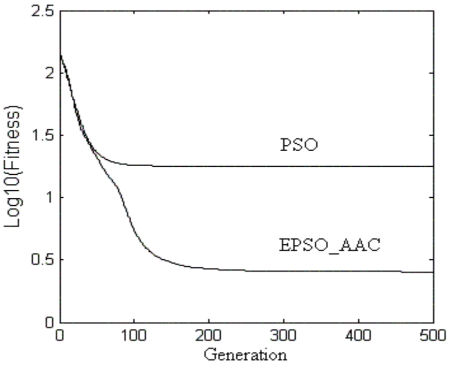


Fig. 4. The convergence curves of *f*3

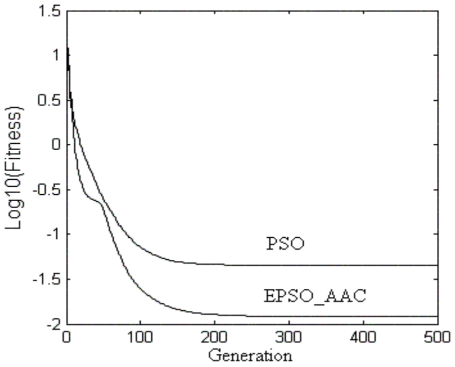


Fig. 5. The convergence curves of *f*4

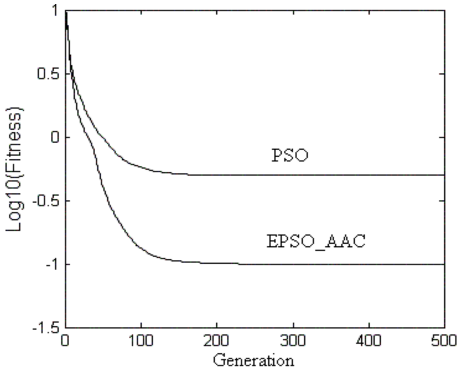


Fig. 6. The convergence curves of *f*5

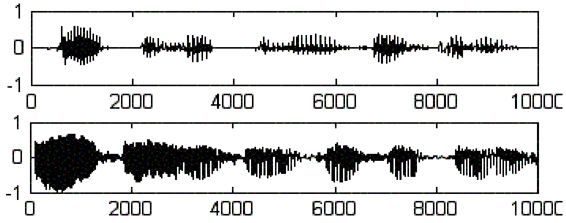


Fig. 7. Source Signals

A computer simulation was conducted to test the EPSO_AAC based algorithm to blind separation of independent sources from their nonlinear mixture. Consider the mixing case of two independent random signals. The mixing matrix A was randomly generated, nonlinear function $f_1(x) = x^3$, $f_2(x) = \tanh(0.5x)$. Nonlinear function $h_1(\cdot)$ and $h_2(\cdot)$ in (11) is selected as $h_1(x)=x^3$, $h_2(x) = x - \tanh(0.5x)$. Sources signals are shown in

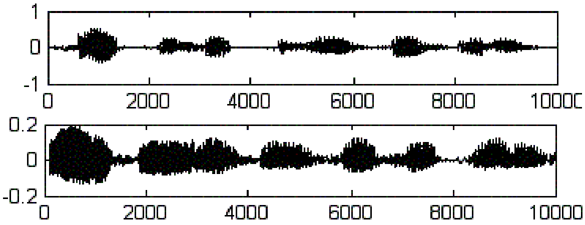


Fig. 8. Separated Signals

Fig.7, the population size was N=50, inertia weight W was 0.9. The maximum generation for EPSO_AAC process to be 1000. Separated signals are shown in Fig.8.

6 Conclusions

In this paper, first, we present an extended particle swarm optimization with adaptive acceleration coefficients (EPSO_AAC). By replacing personal best particle with the average of personal best particles in swarm, more information can be transferred among particles of swarm. And adaptive acceleration coefficients are introduced by establishing a nonlinear functional relationship between acceleration coefficients and different of average fitness and best fitness to avoid premature convergence in the early stages of the search and to enhance convergence to the global optimum solution during the latter stages of the search. Then, EPSO_AAC algorithm is applied to nonlinear blind source separation. A multi-input multi-output B-spline neural network is used to model the demixing system of the nonlinear mixtures, and its weights are optimized under the criterion of independence of its outputs by EPSO_AAC. A set of 5 benchmark function has been used to test EPSO_AAC in comparison with PSO. For most of the benchmark functions, EPSO_AAC found better results than those generated by PSO. and EPSO_AAC is effective to be applied to nonlinear blind source separation.

References

[1] Kennedy J., Eberhart R. Particle swarm optimization. In: IEEE Int'1 Conf. On Neural Networks. Perth, Australia, 1995,1942-1948

[2] Eberhart R. , Kennedy J. A new optimizer using particle swarm theory. In: Proc of the sixth international symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, 39-43

[3] He S, Wu Q H,Wen J Y, *et al.* A Particle Swarm Optimizer with Passive Congregation[J]. Biosystems, 2004, 78: 135-147.

[4] Ratnaweera A, Halgamuge S K,Watson H C. Self-organizing Hierarchical Particle Swarm Optimizer with Time-varying Acceleration Coefficients[J]. IEEE Transactions on Evolutionary Computation, 2004, 8 (3) : 240-255.

[5] Monson C K, Seppi K D. The Kalman Swarm-A New Approach to Particle Motion in Swarm Optimization [A]. Proceedings of the Genetic and Evolutionary Computation Conference [C].Sp ringer, 2004. 140-150.

[6] Van den Bergh F, EngelbrechtA P. A Cooperative Approach to Particle Swarm Optimization[J]. IEEE Transactions on Evolutionary Computation, 2004, 8 (3) : 225-239.

[7] Rodriguez A, Reggia J A. Extending Self-organizing Particle Systems to Problem Solving[J]. ArtificialLife, 2004, 10 (4) :379-395.

[8] Kennedy J, Mendes R. Population Structure and Particle Swarm Performance[A]. Proceedings of the IEEE Congress on Evolutionary Computation[C]. 2002. 1671-1676.

[9] Katare S, Kalos A, West D. A Hybrid Swarm Optimizer for Efficient Parameter Estimation [A]. Proceedings of the IEEE Congress on Evolutionary Computation[C]. 2004. 309-315.

[10] Fan S K S, Liang Y C, Zahara E. Hybrid Simplex Search and Particle Swarm Optimization for the Global Optimization of Multimodal Functions[J]. Engineering Optimization, 2004, 36(4) : 401-418.

[11] Sun J, Feng B, Xu W B. Particle Swarm Optimization with Particles Having Quantum Behavior[A]. Proceedings of the IEEE Congress on Evolutionary Computation[C]. 2004. 325-331.

[12] Taleb A. Jutten C Source separation in post-nonlinear mixtures. IEEE Trans. On Signal Processing, Vol.47,No.10,October 1999. 2807-2820

[13] Y. Tan and J. Wang. Nonlinear blind source separation using Higher order statistical and a genetic algorithm. IEEE Trans. On Evolutionary Computation. Vol.5, No.6, December 2001,600-612

[14] Campolucci, P., Capparelli, F., Guarnieri, S., Piazza, F., Uncini, “A. Neural Networks with Adaptive Spline Activation Function” In Proceedings of IEEE MELECON 96, Bari Italy, 1442-1445. 1996.

[15] Lorenzo Vecci, Francesco Piazza, Aurelio Uncini, "Learning and Approximation Capabilities of Adaptive Spline Activation Function Neural Networks", Neural Networks, Vol.11, No.2, pp 259-270, March 1998.

[16] Papoulis A. Probability, Random Variables, and Stochastic Process. 3rd edition. New York: McGraw-Hill, 1991, 190-191

Application of a Hybrid Ant Colony Optimization for the Multilevel Thresholding in Image Processing

Yun-Chia Liang[1], Angela Hsiang-Ling Chen[2], and Chiuh-Cheng Chyu[1]

[1] Department of Industrial Engineering and Management, Yuan Ze University,
135 Yuan-Tung Road, Chung-Li, Taoyuan County, Taiwan 320, R.O.C.
[2] Department of Financial Management, Nanya Institute of Technology, 414, Sec. 3,
Chung-Shang E. Rd., Chung-Li, Taoyuan County, Taiwan 320, R.O.C.

Abstract. Our study proposes a hybrid optimization scheme based on an ant colony optimization algorithm with the Otsu method to render the optimal thresholding technique more applicable and effective. The properties of discriminate analysis in Otsu's method are to analyze the separability among the gray levels in the image. The ACO-Otsu algorithm, a non-parametric and unsupervised method, is the first-known application of ACO to automatic threshold selection for image segmentation. The experimental results show that the ACO-Otsu efficiently speed up the Otsu's method to a great extent at multi-level thresholding, and that such method can provide better effectiveness at population size of 20 for all given image types at multi-level thresholding in this study.

1 Introduction

Many applications such as document image analysis, map processing, scene processing, computer vision, pattern recognition, and quality inspection of materials consider the image thresholding technique a crucial operation because further process steps have to rely on the segmentation results. The widely-used technique which extracts the objects from the background has both bi-level and multilevel types recognized. For an image with clear objects in the background, the bi-level thresholding which divides the object pixels at one gray level while the background pixels at another is widely used. For rather complex images, on the other hand, the multilevel thresholding segments the pixels into several distinct groups in which the pixels of the same group have gray levels within a specific range. Although recent practices have widely exploited the multilevel thresholding, the complexity of the thresholding problem and the computation time to solve such problem still impose significant challenges as the number of levels required increases. For this reason, many thresholding techniques have been proposed to solve various images segmentation problems and classified by their distinctiveness. For instance, some techniques are identified as either global or local thresholdings based on the role of the intensity value [3, 5, 11, 13, 14] while other methods have been classified as either optimal or property-based [2, 9, 16, 17]. Moreover, Abutaleb [1] classified them into

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1183 – 1192, 2006.

parametric or non-parametric approaches. Parametric approaches assume each group having the probability density function of a Gaussian distribution and find an estimate of the parameters of such distribution which will best fit the given histogram data [12, 13]. Unfortunately, when the desired number of classes is much lower than the number of peaks in the original histogram, the computation time to find the solutions of threshold values often becomes expensive. Different from parametric approaches, non-parametric methods which find the threshold level according to some discriminating criteria are proven to be more computationally efficient and simpler to apply. Examples of non-parametric approaches are such as the entropy [6], cross entropy [8], minimum error [7], and between-class variance [3, 9, 10].

Despite the fact that the problem of thresholding has been quite extensively studied for many years, the automatic determination of an optimum threshold value continues to be of great challenge. Therefore alternative ways to solve multi-level thresholding have been to use heuristics in recent years. Yin [15] proposed a fast scheme for optimal thresholding using genetic algorithms. His method, an optimal thresholding technique, has shown better performance than those of some property-based ones. Zahara *et al.* [18, 19] presented a hybrid optimization scheme which applied the Otsu's method with Nelder-Mead simplex search and particle swarm optimization (the NM-PSO-Otsu method) and proven to not only expedite the Otsu's method efficiently but also extent its effectiveness to a multi-level thresholding problem.

In this paper, a fast scheme using ant colony optimization algorithm is proposed to render the optimal thresholding technique more applicable and effective. We employed the properties of discriminate analysis using Otsu's method to analyze the separability among the gray levels in the image. The remainder of the paper covers: first, a description of the Otsu's method using the concept of discriminate analysis in detail; then, a description of the ant colony optimization (ACO) algorithm and its key features implemented for finding the optimal threshold values; next, a detail of the computational experiments and result comparisons performed to evaluate our algorithm with other methods published; finally, the conclusions of the present work.

2 Otsu's Method for Image Thresholding

The concept of using discriminate analysis for classification problems was first introduced by Fisher [4] and was applied on image thresholding by Otsu [10]. In Otsu's paper, the elementary case of threshold selection where only the gray-level histogram suffices without other a priori knowledge is discussed, and their method is proposed from the viewpoint of discriminate analysis. The feasibility of evaluating the "goodness" of threshold is done through exhaustive search to maximize the between-class variance between dark and bright regions of the image. Our study uses the extended properties of the discriminate criterion to determine the number of objects into which the image should be segmented, and describes the concept of an automatic multilevel thresholding method as follows.

For multi-level thresholding, a gray level image $f(x,y)$ is transformed to a multi-level image $g(x,y)$ by a threshold set $T=\{t_1,t_2,..,t_n,...,t_k\}$, which is composed of k

thresholds. With a given gray level i, n_i denote the observed occurrence frequencies (histogram) of pixels and the total number of pixels $N = n_1 + n_2 + ... + n_L$ where L is the number of gray values in the histogram. Then the gray-level histogram is normalized and regarded as a probability distribution having a given gray level *i*:

$$p_i = \frac{n_i}{N}, \quad p_i \geq 0, \quad \sum_{i=1}^{L} p_i = 1 \tag{1}$$

Suppose we segment these pixels into a suitable number of classes. With *k* denoting the number of selected thresholds (i.e. $0 \leq k \leq L-1$), the image is then partitioned into *k*+1 classes which can be represented by $C_0 = \{0,1,...,t_1\},...,C_n = \{t_n + 1, t_n + 2,...,t_{n+1}\},...,C_k = \{t_k + 1, t_k + 2,...,L-1\}$. Hence, the probabilities of class occurrences (w_n), the class-mean levels (μ_n), and the class variances (σ_n^2) are given as follows, respectively:

$$w_n = \sum_{i=t_n+1}^{t_{n+1}} p_i \text{ , } \mu_n = \frac{\sum_{i=t_n+1}^{t_{n+1}} ip_i}{w_n}, \text{ and } \sigma_n^2 = \frac{\sum_{i=t_n+1}^{t_{n+1}} p_i (i - \mu_n)^2}{w_n} \tag{2}$$

The within-class variance, denoted by σ_{WC}^2, of all segmented classes of pixels is computed as,

$$\sigma_{WC}^2 = \sum_{n=0}^{k} w_n \sigma_n^2 \tag{3}$$

The between-class variance, denoted by σ_{BC}^2, is used to measure the separability among all classes and is expressed as,

$$\sigma_{BC}^2 = \sum_{n=0}^{k} w_n (\mu_n - \mu_T)^2 \tag{4}$$

The total variance σ_T^2 and the overall mean μ_T of pixels in a given gray level image $f(x, y)$ are computed as,

$$\sigma_T^2 = \sum_{i=0}^{L-1} (i - \mu_T)^2 p_i \text{ , and } \mu_T = \sum_{i=0}^{L-1} ip_i \tag{5}$$

In order to evaluate the "goodness" of the threshold at level *k*, the following discriminate criterion measures are used:

$$\lambda = \frac{\sigma_{BC}^2}{\sigma_{WC}^2}, \quad \kappa = \frac{\sigma_T^2}{\sigma_{WC}^2}, \text{ and } \eta = \frac{\sigma_{BC}^2}{\sigma_T^2} \tag{6}$$

Among these measures, the parameter η is the simplest one with respect to *k*, and therefore the optimal threshold k^* that maximizes η, or equivalently maximizes σ_{BC}^2 is also followed as,

$$\sigma_{BC}^2(k_1^*, k_2^*, ..., k_L^*) = \max_{1 \le k_1 < ... < k_L < L} \sigma_{BC}^2(k_1, k_2, ..., k_L) \tag{7}$$

3 Ant Colony Optimization (ACO) Algorithm

Just like other meta-heuristics inspired by the natural process, the Ant Colony Optimization (ACO) algorithm is imitating the behavior of real ants. In ACO, a colony of simple agents, called artificial ants, search for good solutions at every generation. Every artificial ant of a generation builds up a solution based on a state transition probability. Once all ants have their solutions built up, these solutions will be evaluated according to Otsu's method; then, the algorithm will record the best one found so far. The pheromone trails are then updated, and the following ants of the next generation are attracted by the pheromone so that they will likely search nearby these areas. The procedure repeats until the stopping criterion is reached. The ACO algorithm has its general framework like below.

```
 Set all parameters and initialize the pheromone trails
 **Loop** (no. of iterations, *NI*)
   ***Sub-Loop*** (population size, *NA*)
         Build solutions based on **the state transition probability**
   ***Continue*** until all ants have been generated
 Evaluate all solutions during the iteration and select the best one
 Apply the **pheromone update** rule
**Continue** until **the stopping criterion** is reached
```

For threshold selection, the ***state transition probability*** shown below is used in the solution construction process.

$$p_{ij} = \begin{cases} \dfrac{(\tau_{ij})^\alpha}{\sum\limits_{l \in \{1,2,..,UB_i - LB_i + 1\}} (\tau_{il})^\alpha} & j \in \{1,2,..,UB_i - LB_i + 1\} \\ 0 & otherwise \end{cases} \tag{8}$$

where *i* denotes the index of the threshold at multi-level (e.g. *i* = 1 for bi-level, *i* = 2 for tri-level, and *i* = 3 for four-level), *j* refers to the index for the gray level ranging from the pre-specified lower bound and the upper bound of the *i*th threshold, and the lower and upper bound (i.e. LB_i and UB_i) are defined to be the lower and highest value of the gray level index *j*, respectively. In addition, α denotes the parameter controlling the relative weight of pheromone. The pheromone trails, denoted by τ_{ij}

are constantly updated according to the pheromone updating rule. We say, the threshold value with larger pheromone intensity has higher chance to be selected.

After all ants have generated solutions and the best ant so far has been updated, the pheromone update rule, formally expressed as $\tau_{ij}^{new} = \rho \cdot \tau_{ij}^{old} + (1-\rho)\cdot \Delta\tau^{e}$, is performed, where a parameter $\rho \in [0,1]$ controls the pheromone persistence and its $1-\rho$ represents the proportion of the pheromone evaporated. Also, $\Delta\tau^{e}$ denotes the amount of pheromone trail added to τ_{ij} by the best ant for all combinations (*i, j*) belonging to the best solution found so far, and is determined by $\Delta\tau^{e} = Q \times \sigma_{e,BC}^{2}$ where a constant *Q* controls the magnitude of the pheromone contribution, and $\sigma_{e,BC}^{2}$ is the between-class variance of the elitist ant.

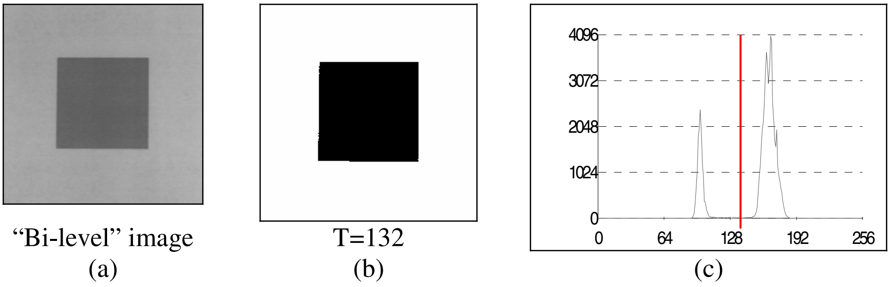
4 Computational Results and Analysis

In this section, the performance of the proposed ACO-Otsu algorithm has been evaluated and compared to the Otsu's method with exhaustive search and the NM-PSO-Otsu algorithm [18, 19]. Our test images were taken under natural room lighting without the support of any special light source and have been transformed into several gray-scale images by thresholding at multi-levels. These images composed of a collection of pixels have been assigned values from 0 to 255, or 1 to 256. Since we began our experiment with the parameter settings of the ACO-Otsu method, we implemented three standard images with rectangular objects of uniform gray values (see Figure 1-a, Figure 2-a, and Figure 3-a). As well as, we employed another three test images – "Dragon", "Screws", and "Blocks" (shown in Figures 4-a, 5-a, and 6-a respectively) to analyze and evaluate the performance of our algorithm. All six images in this study were assigned with uniform gray values at (0~255) range.

Our ACO-Otsu method was implemented on a Pentium IV 3.0GHz, 768 MB personal computer using C++ programming language while the Otsu method with NM-PSO-Otsu methods [18, 19] were implemented on a Athlon XP 2200+ (166×11) with 1 GB RAM using MATLAB®. Both Zahara *et al.* [18, 19] and our studies employed the maximum number of iterations as the termination condition. Our preliminary experiments in Table 1 have shown the following set of all parameters to account for both efficiency and effectiveness; therefore, we set up as follows: $\alpha = 1$, $\tau_0 = 0.01$, $\rho = 0.9$, and $Q = 0.01\tau_0$ for all experimental runs.

Table 1. Settings of different parameters implemented in the ACO-Otsu algorithm

Parameters	Values		
α	0.5	**1**	2
ρ	0.1	0.5	**0.9**
Q	10^{-3}	$\mathbf{10^{-4}}$	10^{-5}
τ_0	10^{-1}	$\mathbf{10^{-2}}$	10^{-3}

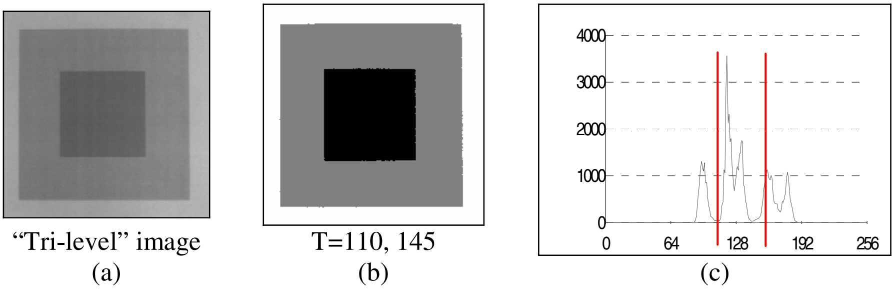


"Bi-level" image
(a)

T=132
(b)

(c)

Fig. 1. Bi-level thresholding test image: (a) original image, (b) ACO-Otsu method, and (c) histogram and the optimal threshold of (b)

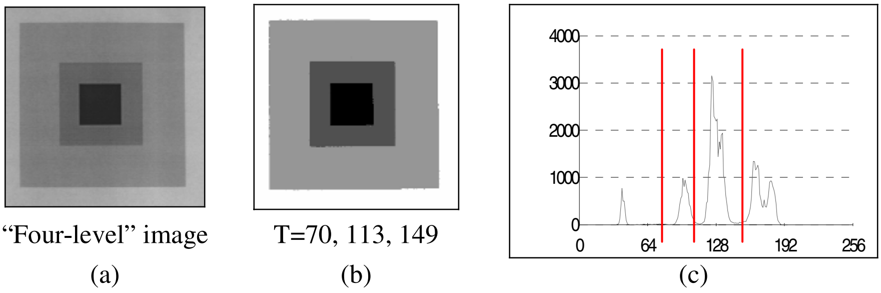


"Tri-level" image
(a)

T=110, 145
(b)

(c)

Fig. 2. Tri-level thresholding test image: (a) original image, (b) ACO-Otsu method, and (c) histogram and the optimal threshold of (b)



"Four-level" image
(a)

T=70, 113, 149
(b)

(c)

Fig. 3. Four-level thresholding test image: (a) original image, (b) ACO-Otsu method, and (c) histogram and the optimal threshold of (b)

Three standard test images (shown in Figures 1-a, 2-a, and 3-a, respectively) are rectangular objects of uniform gray values and the resulting images of the bi-level, tri-level, and four-level (shown in Figures 1-b, 2-b, and 3-b, respectively) verify that ACO-Otsu method can provide a quality performance in image segmentation. Comparison results (see Table 2) for these three standard test images have revealed identical optimal threshold values for both Otsu's and NM-PSO-Otsu methods in [18, 19]. However, our ACO-Otsu method shows slightly different optimal threshold values due to different gray-level scales: (1, 256) in [18, 19] while (0, 255) in this study. Likewise, different objective functions have also been employed for both studies. Zahara *et al.* [18, 19] minimized the within-group variance while our study maximized the between-class variance. As a result, both studies exhibit greatly different optimal objective values (shown in Table 2).

Table 2. Computational results for the three standard test images at the multi-level thresholding

Standard Test Images	Optimal Threshold		Optimal Objective Value (over 10 runs)	
	Otsu and NM-PSO-Otsu	ACO-Otsu	Otsu and NM-PSO-Otsu	ACO-Otsu
Bi-level	133	132	29.85	737.89
Tri-level	111, 146	110, 145	54.11	622.72
Four-level	71, 114, 150	70, 113, 149	36.93	967.70

From Table 3, we can generally conclude that the higher levels of thresholding will cause increasing population sizes and maximum numbers of iterations. As well, CPU times linearly go up when the levels of thresholding and computational complexity go up. Our CPU times vary from the lowest 0.009 seconds at NA=10 and NI=10 (i.e. No. of evaluations = 100) to the highest 0.214 seconds at NA=20 and NI=60 (i.e. No. of evaluations = 1200). When we compare with the Otsu's method, our ACO-Otsu method takes relatively less execution times to achieve 100% optimum. However, the comparison with NM-PSO-Otsu method indicates our ACO-Otsu method performs less efficiently. Such consequence is much expected since our ACO-Otsu method takes the simplest version of ACO as the global optimizer; yet, the NM-PSO-Otsu method employs the hybridization of the local search procedure of Nelder-Mead simplex (NM) and the global optimizer of PSO.

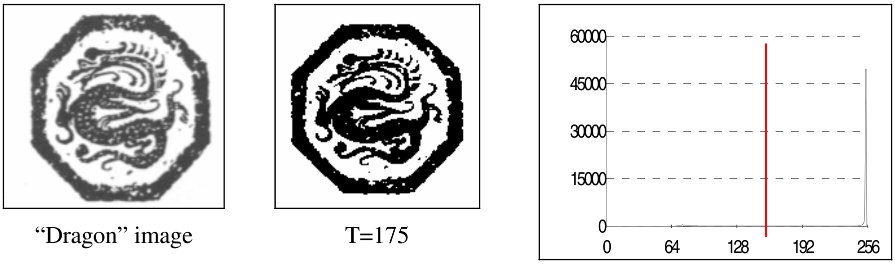
Table 3. Result comparisons among Otsu's, NM-PSO-Otsu and ACO-Otsu methods over the three standard test images

Standard Test Images	CPU Time (sec.)			Population Size (NA) × Max. No. of Iterations (NI)	
	Otsu	NM-PSO-Otsu	ACO-Otsu	Otsu and NM-PSO-Otsu	ACO-Otsu
Bi-level	0.000	0.000	0.009	4 × 10	10 × 10
Tri-level	0.281	0.015	0.044	7 × 20	20 × 20
Four-level	17.828	0.031	0.214	10 × 30	20 × 60

Further to our evaluation on the ACO-Otsu's performance, three images (Dragon, Screws, and Blocks) are chosen; the threshold selection values and computation times for these three tested images are listed in Table 4. ACO-Otsu is able to find the optimal threshold within 0.07 seconds for two-level consideration in all images.

Table 4. Computational results for images of Dragon, Screws, and Blocks

Images	No. of Levels	Optimal Threshold	CPU Time (sec.)	Optimal Objective Value	Population Size (NA) × Max. No. of Iterations (NI)
Dragon	2	175	0.068	2615.11	20 × 20
Screws	2	209	0.042	338.82	20 × 20
	3	194, 226	0.166	393.68	20 × 60
Blocks	2	201	0.009	230.36	5 × 20
	3	196, 228	0.106	274.12	20 × 40

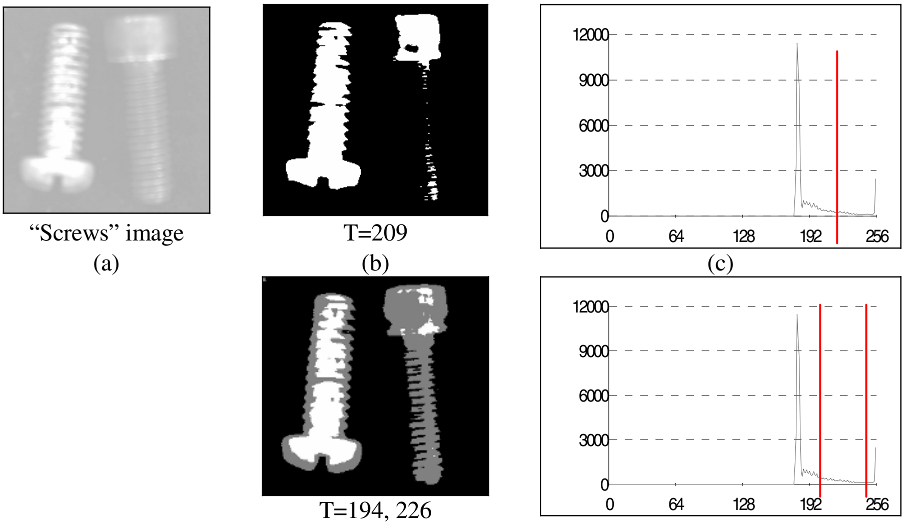


“Dragon” image (a)

T=175 (b)

(c)

Fig. 4. Thresholding result of “Dragon” image: (a) original image, (b) ACO-Otsu method, and (c) histogram and the optimal threshold of (b)



“Screws” image (a)

T=209 (b)

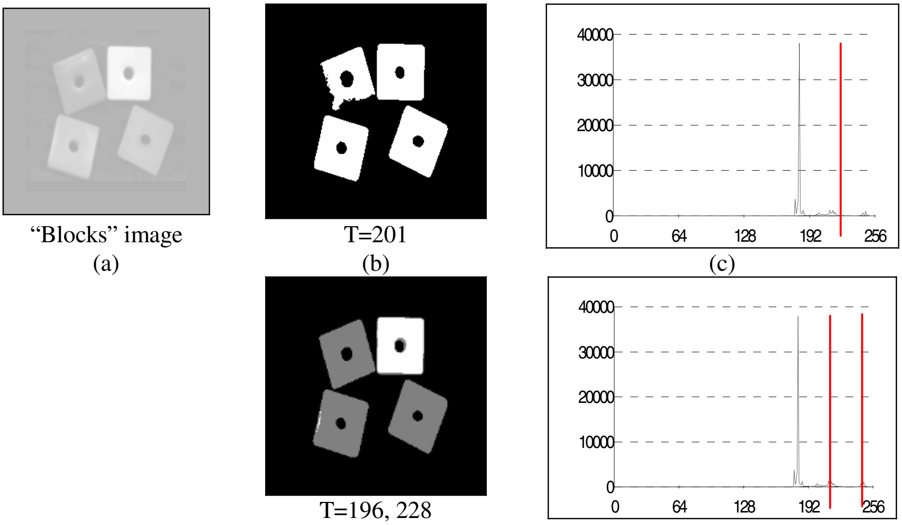
(c)

T=194, 226 (d)

(e)

Fig. 5. Thresholding results of “Screws” image: (a) original image, (b) ACO-Otsu method at bi-level, (c) histogram of (b), (d) ACO-Otsu method at tri-level, and (e) histogram and the optimal threshold of (d)

However, the pictures illustrated in Figures 5-b and 6-b cannot fully detail all objects. For that reason, we add an extra thresholding level to both “Screws” and “Blocks” images. As shown in Table 4, the objective values at tri-level are larger, and both “Screws” and “Blocks” images thresholded at the tri-level exhibit better pictorial results (shown in Figures 5-d and 6-d below) than previous ones.



“Blocks” image
(a)

T=201
(b)

(c)

T=196, 228
(d)

(e)

Fig. 6. Thresholding result of “Blocks” image: (a) original image, (b) ACO-Otsu method at bi-level, (c) histogram and the optimal threshold of (b), (d) ACO-Otsu method at tri-level, and (e) histogram and the optimal threshold of (d)

5 Conclusion

In this study, we can draw several general conclusions at the end of this analysis. First of all, according to our literature reviews in this study, the use of the ACO algorithm is the first application on multi-level thresholding. Second of all, from our preliminary experiments, we also find that when the level of thresholding increases, both NA and NI will increase. Thirdly, the computational results for most images at bi-, tri-, and four-level thresholdings have shown number of iterations (NI) appears to be rather effective at population size (NA) of 20, NA regardless the complexity of the images. Lastly, our ACO-Otsu method out beats the Otsu’s method but not the NM-PSO-Otsu method.

While the quality of image segmentation does not get compromised, we consider the ACO-Otsu method to be a potential method to accelerate the Otsu’s method in multi-level thresholding for real-time applications. However, since our ACO-Otsu method takes the simplest version of ACO as the global optimizer, future investigation for better solution quality and algorithmic efficiency can be done by adding supplementary mechanisms such as local search in ACO algorithm.

References

1. Abutaleb A.S.: Automatic thresholding of gray-level pictures using two-dimensional entropy, Comput. Vis. Graph. Image Process. 47 (1989) 22-32.
2. Belkasim S., Ghazal A., Basir O.A.: Phase-based optimal image thresholding, Digital Signal Process. 13 (2003) 636-655.
3. Cao L. Shi Z.K., Cheng E.K.W.: Fast automatic multilevel thresholding method, Electron. Lett. 38 (2002) 868-870.
4. Fisher R.A.: The use of multiple measurements in taxonomic problems, Ann. Eugenics 7 (1936) 179-188.
5. Jian X., Mojon D.: Adaptive local thresholding by verification-based multithresholding probing with application to vessel detection in retinal images, IEEE Trans. Pattern. Anal. Machine Intell. 25 (2003) 131-137.
6. Kapur J.N., Sahoo P.K., Wong A.K.C.: A new method for gray-level picture thresholding using the entropy of the histogram, Comput. Vis. Graph Image Process. 29 (1985) 273-285.
7. Kittler J., Illingworth J.: Minimum error thresholding, Pattern Recogn. 19 (1986) 41-47.
8. Li C.H., Lee C.K.: Minimum cross entropy thresholding, Pattern Recogn. 26 (1993) 617-625.
9. Liao P.S., Chen T.S., Chung P.C.: A fast algorithm for multilevel thresholding, J. Inf. Sci. and Engineering. 17 (2001) 713-727.
10. Otsu N.: A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1979) 62-66.
11. Rosin P.L., Ioannidis E.: Evaluation of global image thresholding for change detection, Pattern Recogn. Lett. 24 (2003) 2345-2356.
12. Tsai D.M.: A fast thresholding selection procedure for multimodal and unimodal histograms, Pattern Recogn. Lett. 16 (1995) 653-666.
13. Wang Q., Chi Z., Zhao R.: Image thresholding by maximizing of nonfusiness of the 2D grayscale histogram, Comput. Image and Vis. Understanding 85 (2002) 100-116.
14. Yan F., Zhang H., Kube C.R.: A multistage adaptive thresholding method, Pattern Recogn. Lett. 26 (2005) 1183-1191.
15. Yin P.-Y.: A fast scheme for optimal thresholding using genetic algorithms, Signal Process. 72 (1999) 85-95.
16. Yin P.-Y., Chen L.-H.: New method for multilevel thresholding using the symmetry a duality of the histogram, J. Electron. Imag. 2 (1993) 337-344.
17. Yin P.-Y., Chen L.-H.: A fast iterative scheme for multi-level thresholding methods, Signal Process. 60 (1997) 305-313.
18. Zahara E.: A Study of Nelder-Mead Simplex Search Method for Solving Unconstrained and Stochastic Optimization Problems. Ph.D. Dissertation, Yuan Ze University, Taiwan (2003).
19. Zahara E., Fan S.-K. S., Tsai D.M.: Optimal multi-thresholding using a hybrid optimization approach, Pattern Recogn. Lett. 26 (2005) 1082-1095.

Author Index

Acar, Levent II-543
Ahn, ChangWook III-807
Ahn, Hyunchul III-420
Ahn, Tae-Chon III-993, III-1079
Aihara, Kazuyuki I-39
Akhter, Shamim II-430
Akimitsu, Toshio I-49
Akyol, Derya Eren III-553
Alahakoon, Damminda II-814
Aliev, Rafik II-860
Aliev, Rashad II-860
Alsina, Pablo Javier II-159
Alvarez, Mauricio I-747
Amarasiri, Rasika II-814
Amin, M. Ashraful II-430
An, GaoYun II-217
Araujo, Carlos A. Paz de III-1160
Arie, Hiroaki I-387
Arik, Sabri I-570
Arun, JB. I-505
Assaad, Mohammad II-831
Assunção, Edvaldo III-118, III-1131
Atiya, Amir F. II-116, II-764

Bae, Suk Joo II-746
Bai, Guo-Qiang I-554
Bai, Hongliang II-448
Bai, Yaohui I-900
Bao, Zheng I-811
Barman, P.C. II-703
Barrile, Vincenzo II-909
Bascle, B. II-294
Basu, S.K. III-781, III-946
Bayhan, G. Mirac III-553
Becerikli, Yasar III-1105
Benton, Ryan G. II-604
Bernier, O. II-294
Bertoni, Fabiana Cristina III-826
Beşdok, Erkan II-632
Bhatia, A.K. III-781, III-946
Bhaumik, Basabi I-82
Bialowas, Jacek I-90
Bittencourt, Valnaide G. III-21
Blachnik, Marcin III-1028
Bo, Liefeng II-404
Boné, Romuald II-831
Bonner, Anthony J. I-280
Boo, Chang-Jin III-938
Botelho, Silvia III-40
Bouchachia, Abdelhamid I-137
Bouzerdoum, Abdesselam II-207, II-260
Bresolin, Adriano de A. II-159
Brown, Warick II-841
Brückner, Bernd III-251
Burken, John III-684

Cacciola, Matteo II-353, II-909
Cai, Sheng Zhen II-379
Cai, Wei II-713
Cai, Xiongcai II-324
Cai, Xun II-661
Cai, Zixing III-711
Canuto, Anne M.P. I-708
Cao, Fei III-596
Cao, Heng III-702
Cao, Zhiguo III-110
Cao, Zhitong I-642
Cardim, Rodrigo III-1131
Cardot, Hubert II-831
Carneiro, Raphael V. I-427
Carvalho, Aparecido A. III-118
Carvalho, Marcio II-1061
Castillo, M. Dolores del I-237
Chan, Lai-Wan III-400
Chang, Bao Rong II-925, III-478
Chang, Chuan-Wei II-850
Chang, Chuan-Yu II-244
Chang, Hong-Hao II-244
Chao, Ruey-Ming III-312
Chau, Kwok-wing II-1101
Chau, Rowena III-295
Che, Yanqiu II-1071
Chen, Angela Hsiang-Ling II-1183
Chen, Bo I-811
Chen, Bo-Wei I-65
Chen, Ching-Horng I-65
Chen, Chung-Ming II-1079
Chen, Guangju III-518

Chen, Haishan II-236
Chen, Hung-Ching(Justin) III-360
Chen, JunYing II-1148
Chen, Kai-Ju II-60
Chen, Luonan II-952
Chen, Ming I-773
Chen, Ruey-Maw II-1108
Chen, Shi-Huang II-925
Chen, Shifu III-754
Chen, Shu-Heng III-450
Chen, SongCan II-369
Chen, Tai-Liang III-469
Chen, Tianping I-379
Chen, Toly III-581, III-974
Chen, Wanhai II-596
Chen, Wenying III-920
Chen, X.K. I-203
Chen, Xi-Jun III-721
Chen, Xin II-1051
Chen, Xucan II-661
Chen, Xue-wen II-140
Chen, Yi-Wei II-850
Chen, Yuehui III-137, III-209
Chen, Yun Wen II-314
Chen, Zhaoqian III-754
Cheng, Ching-Hsue III-469
Cheng, Jian I-892
Chiang, Chen-Han III-469
Chikagawa, Takeshi II-420
Chiou, Hou-Kai III-605
Cho, Sehyeong III-234
Cho, Sung-Bae III-155, III-892
Cho, Sungzoon I-837, II-21
Cho, Tae-jun III-731
Choi, Gyunghyun II-746
Choi, Hwan-Soo II-964
Choi, Jinhyuk I-437
Choi, Jun Rim III-1206
Chow, Tommy W.S. II-679
Chu, Chee-hung Henry II-604
Chu, Fulei III-920
Chu, Ming-Hui II-850
Chuang, Cheng-Long II-1079
Chun, Myung-Geun II-107
Chung, T.K. I-324
Churan, Jan I-1048
Chyu, Chiuh-Cheng II-1183
Cichocki, Andrzej I-1038, III-92
Çivicioğlu, Pinar II-632
Cottrell, M. II-40
Costa, Jose A.F. III-21
Covacic, Márcio R. III-1131
Cox, Pedro Henrique III-1113
Cui, Gang III-817

D'Amico, Sebastiano II-909
da Silva, Ivan Nunes III-826
Daejeon, Technology II-703
Dai, Hongwei II-1071
Dai, Ruwei II-31
de Carvalho, Aparecido Augusto III-1113, III-1131
de Carvalho, Francisco de A.T. II-50, II-934, III-1012
de Souto, Marcilio C.P. III-21
de Souza, Danilo L. II-729
de Souza, Renata M.C.R. II-50
Deaecto, Grace S. III-118
Demirkol, Askin II-543
Deng, Yanfang I-660
Dhir, Chandra Shekhar I-1133
Dias, Stiven S. I-427
Ding, Feng II-721
Ding, Jundi II-369
Ding, Kang III-535
Ding, Steven X. I-495, I-737
Don, An-Jin II-60
Dong, Huailin II-236
Du, Ding III-702
Du, Ji-Xiang II-80
Duan, Gaoyan II-1090
Duan, Zhuohua III-711
Duch, Włodzisław III-1028
Dutra, Thiago I-708

Eom, Jae-Hong III-30
Er, Meng Joo III-1002
Ertunc, H. Metin III-508

Fan, Ding III-572
Fan, Hou-bin III-322
Fan, Ling II-236
Fan, Shu II-952
Fan, Xuqian I-1098
Fan, Zhi-Gang II-187
Fang, Rui III-692
Fardin, Dijalma Jr. I-427
Farias, Uender C. III-118
Fayed, Hatem II-116
Fazlollahi, Bijan II-860

Fei, Shumin III-664
Feng, Boqin III-498
Féraud, Raphael II-693
Fernández-Redondo, Mercedes I-477, I-616, I-688
Fischer, T. II-40
Freeman, Walter J. I-11
Fu, Chaojin I-591
Fu, Hui III-268
Fu, Zhumu III-664
Fujimoto, Katsuhito II-88
Fujita, Kazuhisa I-255, I-263
Funaya, Hiroyuki III-746
Fung, Chun Che II-841
Furukawa, Tetsuo I-935, I-943, I-950, I-958, II-278
Fyfe, Colin I-361

Gaino, Ruberlei III-118
Gao, Cao III-322
Gao, Cunchen I-598
Gao, Fang III-817
Gao, Fei III-772
Gao, Jianpo II-270
Gao, Shao-xia III-322
Gao, Shihai II-515
Gao, Ying II-1174
Garcez, Artur S. d'Avila I-427
Garg, Akhil R. I-82
Gedeon, Tamás II-841
Geweniger, T. II-40
Gomes, Gecynalda Soares S. II-737
Gong, Jianhua II-890
Gong, Kefei I-978
Gong, Qin I-272
Greco, Antonino II-353, II-909
Gruber, Peter I-1048
Grzyb, Beata I-90
Gu, Ren-Min II-253
Gu, Weikang II-1128
Gu, Wenjin III-702
Guan, Di II-596
Guan, Jing-Hua II-1164
Guan, Tian I-272
Guerreiro, Ana M.G. III-1160
Gui, Chao II-506
Guirimov, Babek II-860
Guo, Bing III-1189
Guo, G. I-203
Guo, Jun I-827
Guo, Meng-shu II-1032
Guo, Ping II-286
Guo, Qiang II-596
Guo, Shen-Bo II-80
Guo, X.C. II-1138
Guo, Yi-nan I-892

Halgamuge, Saman K. I-915, III-260, III-526
Hammer B. II-40
Han, Jialing III-225
Han, Seung-Soo III-234
Hao, Jingbo III-866
Hao, Zhifeng I-773
Hashem, Sherif II-116
Hashimoto, Hideki III-856, III-874
Hattori, Motonobu I-117
He, Bo III-1055
He, Pilian I-978
He, Zhaoshui I-1038
He, Zhenya I-856
Henao, Ricardo I-747
Herbarth, Olf III-278
Hernández-Espinosa, Carlos I-477, I-616, I-688
Hirayama, Daisuke I-255
Hirooka, Seiichi I-263
Hirose, Akira I-49
Ho, Kevin I-521
Hong, Chuleui II-1118
Hong, Jin-Hyuk III-155, III-892
Hong, Jin Keun III-1122
Honma, Shun'ichi II-420
Horiike, Fumiaki II-420
Horio, Keiichi I-907, I-968, III-1168
Hosaka, Ryosuke I-39
Hoshino, Masaharu I-907
Hosino, Tikara I-407
Hotta, Yoshinobu II-88
Hou, Gang III-225
Hou, Xiaodi I-127
Hou, Yuexian I-978
Hou, Zeng-Guang III-721
Hsieh, Kun-Lin III-48
Hsieh, Sheng-Ta I-1078
Hsu, Arthur I-915, III-260, III-526
Hu, Changhua I-718
Hu, Meng I-11
Hu, Wei II-448
Hu, Xiaolin II-994

Hu, Yun-an II-1022
Huang, D. II-679
Huang, De-Shuang II-80
Huang, Jau-Chi I-183, I-1012
Huang, Jie II-481
Huang, K. I-203
Huang, Kaizhu II-88
Huang, Kou-Yuan II-60
Huang, Lijuan I-1022
Huang, Liyu II-533, III-58
Huang, Qinghua I-1058
Huang, Shian-Chang III-390
Huang, Tingwen III-1070
Huang, Wei III-380
Huang, Yueh-Min II-1108
Huang, Zhi-Kai II-80
Hüper, Knut I-1068
Hwang, Hyung-Soo III-993
Hwang, Kyu-Baek I-670
Hwang, Soochan III-341

Ikeda, Kazushi III-746
Ikeguchi, Tohru I-39
In, Myung Ho I-1088
Inuso, G. III-82
Iqbal, Nadeem II-703
Ishii, Kazuo I-935
Ishizaki, Shun I-117
Islam, Md. Atiqul II-430
Itoh, Katsuyoshi III-563
Izworski, Andrzej I-211

Jang, Kyung-Won III-1079
Jeng, Cheng-Chang III-48
Jeong, Sungmoon II-466
Ji, Jian II-387, II-394
Jia, Yinshan I-819
Jia, Yunde III-268
Jian, Jigui III-674
Jiang, Huilan II-124
Jiang, Jiayan II-278
Jiang, L. III-589
Jiang, Minghu III-285
Jiang, Ning I-495
Jiang, Qi II-499
Jiang, Weijin II-870
Jianhong, Chen III-572
Jianjun, Li III-572
Jiao, Licheng II-404
Jin, Dongming III-1199
Jin, TaeSeok III-856, III-874
Jin, Wuyin I-59
Jordaan, Jaco II-974
Ju, Fang II-499
Juang, Jih-Gau III-605, III-654
Jun, Sung-Hae I-864
Jung, Chai-Yeoung III-789
Jung, Soonyoung III-331

Kabe, Takahiro III-1141
Kacalak, Wojciech I-298
Kalra, Prem K. I-608
Kambara, Takeshi I-255
Kamimura, Ryotaro I-626, I-925, II-897
Kang, Min-Jae II-1014, III-938
Kang, Pilsung I-837
Kang, Rae-Goo III-789
Kang, Yuan II-850
Kao, Yonggui I-598
Karabıyık, Ali III-1095
Karadeniz, Tayfun II-824
Karim, M.A. III-526
Kashimori, Yoshiki I-255, I-263
Katsumata, Naoto II-420
Kawana, Akio I-547
Keck, Ingo R. I-1048
Khan, Farrukh Aslam II-1014
Khan, Usman II-651
Kil, Rhee Man I-755
Kim, Byoung-Hee I-670
Kim, Byung-Joo III-192
Kim, C.H. III-964
Kim, Choong-Myung I-290
Kim, Dong Hwee I-290
Kim, Eun Ju II-489, III-350
Kim, H.J. III-964
Kim, Harksoo II-150
Kim, Ho-Chan II-1014, III-938
Kim, Ho-Joon II-177
Kim, HyungJun II-917
Kim, Il Kon III-192
Kim, Kwang-Baek II-167
Kim, Kyoung-jae III-420
Kim, Myung Won II-489, III-350, III-797
Kim, S.S. III-964
Kim, Saejoon I-371
Kim, SangJoo III-856
Kim, Sungcheol II-1118
Kim, Sungshin II-167

Kim, Tae-Seong I-1088
Kim, Yonghwan III-341
Kinane, Andrew III-1178
King, Irwin I-342
Kiong, Loo Chu II-70
Kitajima, Ryozo I-626
Koh, Stephen C.L. III-145
Kong, Feng III-1038, III-1046
Kong, Hui I-447
Kong, Jun III-225
Koo, Imhoi I-755
Koyama, Ryohei I-698
Kraipeerapun, Pawalai II-841
Kropotov, Dmitry I-727
Kui-Dai III-217
Kung, Sun-Yuan I-314
Kuo, Yen-Ting I-1012
Kurashige, Hiroki I-19
Kurogi, Shuichi I-698, III-563
Kuwahara, Daisuke III-563
Kwok, James T. III-11
Kwon, Man-Jun II-107
Kwon, YouAn III-331

La Foresta, F. III-82
Lai, Edmund Ming-Kit I-155, III-370
Lai, Hung-Lin II-60
Lai, Kin Keung III-380, III-928
Lai, Weng Kin II-776
Lam, James I-332
Lang, Elmar W. I-1048
Larkin, Daniel III-1178
Lassez, Jean-Louis II-824
Lee, Dae-Jong II-107
Lee, Geehyuk I-437
Lee, Hyoung-joo II-21
Lee, Hyunjung II-150
Lee, Juho II-177
Lee, Kan-Yuan I-1078
Lee, Kichun III-420
Lee, Kin-Hong II-796
Lee, Kwan-Houng III-731
Lee, Minho II-466
Lee, Sang-joon II-1014
Lee, SeungGwan I-487
Lee, Soo Yeol I-1088
Lee, Soo-Young I-1133, II-703
Lee, Sungyoung I-1088
Lee, Young-Koo I-1088
Leen, Gayle I-361
Lemaire, Vincent II-294, II-693
Leng, Yan II-227
Leung, Chi-sing I-521
Leung, Kwong-Sak II-796
Li, Bo II-458
Li, Bo Yu II-314
Li, Chao-feng II-132
Li, Cheng Hua III-302
Li, Donghai II-515
Li, F. III-589
Li, G. II-586
Li, Guang I-11
Li, H. I-203
Li, Hongwei I-1107
Li, Jianyu I-987
Li, Jiaojie I-11
Li, Jie II-523
Li, Jing II-1022
Li, Kun-cheng I-175
Li, Li II-99
Li, Min II-713
Li, Ping I-737
Li, Qian-Mu III-545
Li, Rui I-1107
Li, Shutao III-11
Li, Sikun II-661
Li, Weihua III-535
Li, Wenye II-796
Li, Xiaobin II-474
Li, Xiaohe I-513
Li, Xiaoli III-66
Li, Xiaolu I-1125
Li, Xiao Ping I-782
Li, Yangmin II-1051
Li, Zhaohui II-1174
Li, Zheng II-596
Li, Zhengxue I-562
Li, Zhishu III-1189
Li, Ziling I-660
Liang, Pei-Ji I-30
Liang, Xun III-410
Liang, Y.C. II-1138
Liang, Yun-Chia II-1183
Liao, Chen III-11
Liao, Guanglan I-1030
Liao, Ling-Zhi I-193, II-343
Liao, Shasha III-285
Liao, Wudai I-529
Liao, Xiaoxin I-529
Lim, Chee Peng II-776

Lim, HeuiSeok III-331
Lim, Heuiseok I-247
Lim, Sungsoo III-892
Lim, Sungwoo II-651
Lin, Chun-Ling I-1078
Lin, Chun-Nan III-48
Lin, Frank I-765
Lin, L. II-586
Lin, Lili II-1128, III-763
Lin, Lizong III-702
Lin, Pan II-379
Lin, Xiaohong II-870
Lin, Y. I-203
Lin, Yao III-498
Ling, Ping I-801
Liou, Cheng-Yuan I-183, I-1012
Liu, Ben-yu II-942
Liu, Bingjie I-718
Liu, Chan-Cheng I-1078
Liu, Changping II-448
Liu, Chunbo II-880
Liu, Da-You II-1164
Liu, Fan-Xin III-461
Liu, Fan-Yong III-461
Liu, Fang II-880
Liu, Feng-Yu III-545
Liu, H. III-589
Liu, Hanxing III-209
Liu, Hongwei I-811, III-817
Liu, Hongyan III-1038, III-1046
Liu, Huailiang II-1174
Liu, Jing II-1042, II-1156
Liu, Li II-880
Liu, Qihe I-1098
Liu, Qingshan II-1004
Liu, Shiyuan I-1030
Liu, Si-Pei II-1164
Liu, Tangsheng II-124
Liu, Wei II-481
Liu, Wen-Kai III-654
Liu, Xiabi III-268
Liu, Xue I-30
Liu, Yanheng III-201
Liu, Yunfeng III-596
Liu, Zhi-Qiang II-671
Lo, C.F. I-324
Lo, Shih-Tang II-1108
Long, Chong II-88
Long, Fei II-236, III-664
Long, Zhi-ying I-175
Loy, Chen Change II-776
Lu, Bao-Liang II-187
Lu, Chiu-Ping I-65
Lu, Huaxiang III-488
Lu, Jie I-175
Lu, Yinghua III-225
Lu, Li I-1030
Lu, Wenlian I-379
Lu, Yinghua III-285
Ludermir, Teresa B. II-737, II-934, II-1061, III-884
Lunga, Dalton III-440
Luo, Dijun II-1
Luo, Jianguo I-900
Luo, Si-Wei I-193, II-343
Luo, Siwei I-987, III-692
Luo, Yinghong I-1117
Lursinsap, Chidchanok I-765

Ma, Aidong III-692
Ma, RuNing II-369
Ma, Run-Nian I-554
Ma, Weimin III-1022
Ma, Xiaoyan III-488
Ma, Xin II-499
Ma, Yunfeng III-1055
Madabhushi, Anant III-165
Magdon-Ismail, Malik III-360
Maia, André Luis S. II-934
Majewski, Maciej I-298
Mak, Man-Wai I-314
Mammone, N. III-82
Man, Kim-Fung II-568
Mani, V. III-964
Mao, Chengxiong II-952
Mao, Ye III-498
Martín-Merino, Manuel I-995
Marwala, Tshilidzi III-430, III-440, III-684, III-1087
Mata, Wilson da II-729
Matsuyama, Yasuo II-420
Meng, Fankun II-515
Meng, Ling-Fu I-65
Meng, Qing-lei II-458
Miao, Dong III-596
Miao, Sheng II-942
Miki, Tsutomu I-352
Mimata, Mitsuru III-563
Ming, Qinghe I-598
Minku, Fernanda L. III-884

Minohara, Takashi II-786
Mishra, Deepak I-608
Mizushima, Fuminori I-228
Mogi, Ken I-147
Molter, Colin I-1
Moon, Jaekyoung II-466
Morabito, Francesco Carlo II-353, II-909, III-82
Morisawa, Hidetaka I-255
Morris, Quaid I-280
Mu, Shaomin I-634, III-184
Mukkamala, Srinivas II-824
Murashima, Sadayuki I-537
Mursalin, Tamnun E. II-430

Nakagawa, Masahiro I-397
Nakajima, Shinichi I-650
Nakamura, Tomohiro II-420
Nam, KiChun I-247, I-290, III-331
Namikawa, Jun I-387
Naoi, Satoshi II-88
Navet, Nicolas III-450
Neto, Adrião Duarte D. II-159, II-729
Ng, G.S. III-145
Ng, S.C. III-165
Nguyen, Ha-Nam I-792, III-1
Nishi, Tetsuo I-827
Nishida, Shuhei I-935
Nishida, Takeshi I-698, III-563
Nishiyama, Yu I-417
Niu, Yanmin II-197

O'Connor, Noel III-1178
Ogata, Tetsuya I-387
Oh, Kyung-Whan I-864
Oh, Sanghoun III-807
Oh, Sung-Kwun III-1079
Ohashi, Fuminori II-420
Ohn, Syng-Yup I-792, III-1
Okabe, Yoichi I-49
Oki, Nobuo III-1131
Oliveira, Hallysson I-427
Ong, Chong Jin I-782
Orman, Zeynep I-570
Oshime, Tetsunari I-1004

Pan, Li II-99
Panchal, Rinku III-127
Park, Changsu I-247
Park, Dong-Chul II-439, II-641, II-964
Park, Ho-Sung III-1079
Park, Hyung-Min I-1133
Park, KiNam III-331
Park, Kyeongmo II-1118
Park, Soon Choel III-302
Parrillo, Francesco II-909
Patel, Pretesh B. III-430
Pei, Wenjiang I-856
Pei, Zheng I-882
Peng, Bo III-110
Peng, Hong I-882
Peng, Hui I-457
Peng, Lizhi III-209
Peng, Wendong II-622
Peng, Xiang I-342
Peng, Yunhui III-596
Phung, S.L. II-207
Pi, Daoying I-495
Piao, Cheng-Ri III-234
Poh, Chen Li II-70
Ptashko, Nikita I-727
Puntonet, Carlos G. I-1048

Qian, Jian-sheng I-892
Qiang, Xiaohu I-1117
Qiao, X.Y. II-586
Qin, Sheng-Feng II-651
Qin, Zheng II-1148
Qiu, Fang-Peng II-880
Quan, Zhong-Hua II-80
Quek, Chai I-155, III-145, III-370

Raicharoen, Thanapant I-765
Rajapakse, Jagath C. II-361, III-102, III-983
Ramakrishnan, A.G. II-361
Ramakrishna, R.S. III-807
Rao, M.V.C. II-70
Rasheed, Tahir I-1088
Ren, Guang III-616
Ren, Guoqiao I-847
Ren, Mingming III-1055
Roeder, Stefan III-278
Roh, Seok-Beom III-993
Rolle-Kampczyk, Ulrike III-278
Román, Jesus I-995
Rosen, Alan I-105
Rosen, David B. I-105
Ruan, QiuQi II-217
Rui, Zhiyuan I-59
Ryu, Joung Woo II-489, III-797

Saad, David II-754
Saeki, Takashi I-352
Sahin, Suhap III-1105
Saito, Toshimichi I-1004, III-738, III-1141
Sakai, Ko I-219
Sakai, Yutaka I-19
Salihoglu, Utku I-1
Samura, Toshikazu I-117
Saranathan, Manojkumar II-361
Sato, Naoyuki I-1
Savran, Aydoğan III-1095
Schleif, F.-M. II-40
Senf, Alexander II-140
Seo, Jungyun II-150
Serrano, J. Ignacio I-237
Shao, Bo I-642
Shao, Hongmei I-562
Shao, Zhuangfeng I-773
Shen, Hao I-1068
Shen, Ji-hong I-467
Shen, Kai Quan I-782
Shen, Lincheng III-900
Shen, Yan III-1189
Shen, Yanjun III-674
Shen, Z.W. I-203
Shenouda, Emad Andrews I-280
Shi, Chaojian II-334
Shi, Tielin III-535
Shieh, Grace S. II-1079
Shu, Zhan I-332
Shukla, Amit I-608
Shyr, Wen-Jye III-954
Silva, Fabio C.D. II-50
Silva, Joyce Q. II-50
Silveira, Patricia III-40
Simas, Gisele III-40
Singare, Sekou II-533, III-58
Sinha, Neelam II-361
Smith, A.J.R. III-526
Smith-Miles, Kate II-814, III-295
Smola, Alexander J. I-1068
Soares, Fola III-684
Son, YoungDae III-874
Song, Baoquan II-612
Song, Tao II-890
Song, Wang-cheol II-1014
Song, Zhihuan I-737
Souto, Marcilo C.P. de I-708
Souza, Alberto F. De I-427
Sowmya, Arcot II-324
Su, Jianbo II-622
Sugano, Shigeki I-387
Sui, Jianghua III-616
Suk, Jung-Hee III-1206
Sum, John I-521
Sun, Baolin II-506
Sun, Jiancheng I-900
Sun, Jun II-88, II-1042, II-1156
Sun, Lei II-304
Sun, Ming I-467, II-1032
Sun, Tsung-Ying I-1078
Sun, Wei II-984
Sun, Zonghai I-874
Suzuki, Satoshi III-738

Tadeusiewicz, Ryszard I-211
Takahashi, Norikazu I-827
Takamatsu, Shingo I-650
Tamukoh, Hakaru III-1168
Tan, Beihai I-1125
Tan, Min III-721
Tan, T.Z. III-145
Tan, Zheng II-713
Tanabe, Fumiko I-147
Tanaka, Shinya I-698
Tanaka, Takahiro I-968
Tang, Zheng II-1071
Tani, Jun I-387
Tao, Caixia I-1117
Teddy, Sintiani Dewi I-155, III-370
Teixeira, Marcelo C.M. III-118, III-1131
Terchi, Abdelaziz II-651
Tettey, Thando III-1087
Theis, Fabian J. I-1048
Tian, Daxin III-201
Tian, Jing III-900
Tian, Jun II-721
Tian, Mei I-193, II-343
Tian, Shengfeng I-634, III-184
Tian, Zheng II-387, II-394, II-474
Tivive, Fok Hing Chi II-260
Tokunaga, Kazuhiro I-943, I-958
Tolambiya, Arvind I-608
Tomisaki, Hiroaki III-563
Tong, Hengqing I-457, I-660, III-772
Tong, Xiaofeng II-448
Torikai, Hiroyuki I-1004, III-1141
Torres-Huitzil, César III-1150

Torres-Sospedra, Joaquín I-477, I-616, I-688
Toyoshima, Takashi I-228
Tran, Chung Nguyen II-439, II-964
Tsai, Hsiu Fen II-925, III-478
Tsukada, Minoru I-72
Tsuzuki, Shinya I-547
Tu, Kai-Ti III-654
Tu, Yung-Chin II-244

Ukil, Abhisek II-578, II-974

Vajpai, Jayashri I-505
Vasiliev, Oleg I-727
Verma, Brijesh III-127
Versaci, Mario II-353
Vetrov, Dmitry I-727
Vialatte, François B. III-92
Villmann, Th. II-40

Wagatsuma, Nobuhiko I-219
Wan, Guojin II-560
Wang, Bin II-334
Wang, Bo II-369
Wang, C.Y. II-1138
Wang, Chi-Shun III-312
Wang, Chunheng II-31
Wang, Cong III-165
Wang, Daojun II-890
Wang, Dianhui III-1189
Wang, Dingwei III-836
Wang, Dongyun I-529
Wang, Fasong I-1107
Wang, Feng II-515
Wang, Gang III-1063
Wang, Guang-Li I-30
Wang, Haiqing I-495, I-737
Wang, Hongfeng III-836
Wang, Hsing-Wen III-390
Wang, Huiyan III-176
Wang, Huiyuan II-227
Wang, J.Z. III-589
Wang, Jian III-201
Wang, Jian-Gang I-447
Wang, Jie III-176
Wang, Jing-Xin III-217
Wang, Jingmin I-847
Wang, Jun II-994, II-1004
Wang, Keyun III-920
Wang, Liang III-692
Wang, Liansheng II-661
Wang, Limin I-680
Wang, Lin III-285
Wang, Ling II-404
Wang, Mengbin II-124
Wang, Peng III-1199
Wang, R.C. III-589
Wang, Rubin I-306
Wang, Shi-tong II-132
Wang, Shouyang III-380, III-928
Wang, Tao II-304, II-412, II-448
Wang, Weirong II-533, III-58
Wang, Xiaodong III-910
Wang, Xin III-636
Wang, Xuchu II-197
Wang, Yan III-137
Wang, Yea-Ping II-850
Wang, Yi II-671
Wang, Yujian II-270
Wang, Yumei I-819
Wang, Zhe I-801
Wang, Zengfeng II-227
Wang, Zheng-you II-132
Wang, Zhengzhi II-612
Wang, Zhi-Ying III-217
Wang, Zhongsheng I-591
Watanabe, Kazuho I-407
Watanabe, Osamu I-165, I-547
Watanabe, Sumio I-407, I-417, I-650
Wei, Pengcheng III-243
Wei, Yaobing I-59
Wen, Wen I-773
Weng, ZuMao II-379
Wilder-Smith, Einar P.V. I-782
Wong, K.Y. Michael II-754
Wong, Kok Wai II-841
Woo, Dong-Min II-641
Woodley, Robert S. II-543
Wright, David II-651
Wu, C.G. II-1138
Wu, Jianhua II-560
Wu, Lu I-598
Wu, M. III-589
Wu, R.H. I-203
Wu, Wei I-562
Wu, Xia I-175
Wu, Xiaojuan II-227
Wu, Xiaopei III-74
Wu, Xiu-Ling II-523
Wu, Yan II-253

Wu, Yiqiang II-560
Wu, Yunfeng III-165
Wu, Zhenyang II-270

Xi, Lixia II-1090
Xia, Chang-Liang III-626, III-645
Xiao, Baihua II-31
Xiao, Mei-ling II-942
Xie, Shengli I-1038, II-806
Xiong, Li I-457
Xiong, Rong II-1
Xiu, Jie III-626, III-645
Xu, Anbang II-286
Xu, Lei II-31
Xu, Man-Wu III-545
Xu, Qing II-304
Xu, Qinzhen I-856
Xu, Wenbo II-1042, II-1156
Xu, Yao-qun I-467, II-1032
Xu, Yuetong I-642
Xu, Yulin I-529
Xuan, Jianping I-1030

Yamaguchi, Nobuhiko II-11
Yamaguchi, Yoko I-1
Yamakawa, Takeshi I-907, I-968, III-1168
Yamazaki, Yoshiyuki I-72
Yan, Changfeng I-59
Yan, Shaoze III-920
Yang, Bo III-137, III-209
Yang, Bojun II-1090
Yang, Chao III-1055
Yang, Chenguang II-984
Yang, Degang III-243
Yang, Huaqian III-243
Yang, Hui III-636
Yang, Hui-zhong II-721
Yang, Hyun-Seung II-177
Yang, I-Ching III-48
Yang, Jian III-410
Yang, Jie I-1058
Yang, Jihua III-1063
Yang, Jilin I-882
Yang, Jun-Jie II-880
Yang, Luxi I-856
Yang, Shuzhong I-987
Yang, Wen-Chie I-183
Yang, Xiaogang III-596
Yang, Xiaowei I-773
Yang, Yu II-1071
Yang, Yulong III-225
Yang, Zhiyong III-702
Yao, Chun-lian II-458
Yao, Li I-97, I-175
Yao, Wangshu III-754
Yazici, Suleyman III-1105
Ye, Datian I-272
Ye, Liao-yuan II-942
Ye, Mao I-1098, III-498
Ye, Zhongfu III-74
Yeh, Chung-Hsing III-295
Yeo, Jiyoung II-466
Yeung, C.H. II-754
Yeung, Sai-Ho II-568
Yim, Hyungwook I-247
Yin, Chuanhuan I-634, III-184
Yin, Hujun II-671
Yin, Jianping III-866
Yin, Kai I-97
Yin, Yuhai III-702
Yoon, PalJoo II-466
Yoshida, Fumihiko I-626, II-897
You, Jiun-De II-60
Youn, Jin-Seon III-1206
Yu, Gang II-379
Yu, Gisoon I-290
Yu, Hui III-674
Yu, Jinxia III-711
Yu, Lean III-380, III-928
Yu, Li II-1090
Yu, Shi III-572
Yu, Xuegang III-201
Yu, Zhongyuan II-1090
Yuan, Haiying III-518
Yuan, Runzhang III-209
Yuan, Xu-dong III-322
Yuan, Zhanhui III-1063
Yumak, Onder III-508

Zeng, Delu II-806
Zeng, Xiangyan I-1117
Zhang, Anne II-140
Zhang, Boyun III-866
Zhang, Byoung-Tak I-670, III-30
Zhang, Changjiang III-910
Zhang, Haisheng III-410
Zhang, Hong III-545
Zhang, Huanshui I-580
Zhang, Jiadong II-952

Zhang, Jun-ben II-132
Zhang, Ke I-537, III-845
Zhang, Kun III-400
Zhang, Liming II-278
Zhang, Liqing I-127, II-523
Zhang, Pu-Ming I-30
Zhang, Taiyi I-513
Zhang, Wei III-243
Zhang, Xiaoguang II-1090
Zhang, Xin II-612
Zhang, Xingzhou II-596
Zhang, Xun III-1199
Zhang, Yan II-304, II-412
Zhang, Ye II-560
Zhang, Yimin II-448
Zhang, Y.P. I-203
Zhang, Yong Sheng II-412
Zhang, Yun-Chu III-721
Zhang, Zhi-Lin II-523
Zhang, Zhikang I-306
Zhao, Chuang II-553
Zhao, Lian-Wei I-193, II-343
Zhao, Qibin II-523
Zhao, Weirui I-580
Zhao, Xiao-Jie I-97
Zhao, Yibiao III-692
Zhao, Yongjun II-481, II-553
Zheng, Bo III-102
Zheng, Hong II-99
Zheng, Hui I-782, II-1174
Zheng, Shiyou III-664
Zheng, XiaoJian II-379
Zheng Zhai, Yu III-260
Zhong, Yixin III-165
Zhou, Chengxiong III-928
Zhou, Chunguang I-801
Zhou, Jian-Zhong II-880
Zhou, Juan II-361, III-983
Zhou, Li-Zhu II-671
Zhou, Weidong II-499
Zhou, Wenhui II-1128
Zhou, Yatong I-513, I-900
Zhou, Yi III-1002
Zhou, Yue I-1058
Zhou, Zhiheng II-806
Zhu, Jun II-890
Zhu, Li III-498
Zhu, Xiaoyan II-88
Zhuang, Li II-88
Zou, An-Min III-721
Zuo, Bin II-1022