

Chi Sing Leung
Minho Lee
Jonathan H. Chan (Eds.)

LNCS 5863

Neural Information Processing

16th International Conference, ICONIP 2009
Bangkok, Thailand, December 2009
Proceedings, Part I

1
Part I

 Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany

Chi Sing Leung Minho Lee
Jonathan H. Chan (Eds.)

Neural Information Processing

16th International Conference, ICONIP 2009
Bangkok, Thailand, December 1-5, 2009
Proceedings, Part I

Volume Editors

Chi Sing Leung
City University of Hong Kong
Department of Electronic Engineering
Hong Kong
E-mail: eeleungc@cityu.edu.hk

Minho Lee
Kyungpook National University
School of Electrical Engineering and Computer Science
1370 Sankyuk-Dong, Puk-Gu, Taegu, 702-701, Korea
E-mail: mhlee@knu.ac.kr

Jonathan H. Chan
King Mongkut's University of Technology Thonburi
School of Information Technology
126 Pracha-U-Thit Rd., Bangmod, Thungkru, Bangkok 10140, Thailand
E-mail: jonathan@sit.kmutt.ac.th

Library of Congress Control Number: 2009939833

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

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

ISSN 0302-9743
ISBN-10 3-642-10676-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-10676-7 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

springer.com

© Springer-Verlag Berlin Heidelberg 2009
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12799459 06/3180 5 4 3 2 1 0

Preface

This two-volume set constitutes the Proceedings of the 16th International Conference on Neural Information Processing (ICONIP 2009), held in Bangkok, Thailand, during December 1–5, 2009. ICONIP is a world-renowned international conference that is held annually in the Asia-Pacific region. This prestigious event is sponsored by the Asia Pacific Neural Network Assembly (APNNA), and it has provided an annual forum for international researchers to exchange the latest ideas and advances in neural networks and related discipline. The School of Information Technology (SIT) at King Mongkut's University of Technology Thonburi (KMUTT), Bangkok, Thailand was the proud host of ICONIP 2009. The conference theme was "Challenges and Trends of Neural Information Processing," with an aim to discuss the past, present, and future challenges and trends in the field of neural information processing.

ICONIP 2009 accepted 145 regular session papers and 53 special session papers from a total of 466 submissions received on the Springer Online Conference Service (OCS) system. The authors of accepted papers alone covered 36 countries and regions worldwide and there are over 500 authors in these proceedings. The technical sessions were divided into 23 topical categories, including 9 special sessions. Technical highlights included a keynote speech by Shun-ichi Amari (the founder of APNNA); plenary and invited talks by Włodzisław Duch (President of the European Neural Network Society), Kunihiko Fukushima, Tom Gedeon, Yuzo Hirai (President of the Japanese Neural Network Society), Masumi Ishikawa, Nikola Kasabov (President of the International Neural Network Society), Minh Lee, Soo-Young Lee, Andrew Chi-Sing Leung, Bao-Liang Lu, Chidchanok Lursinsap, Paul Shaoning Pang, Ron Sun, Shiro Usui, DeLiang Wang, Jun Wang, Lipo Wang and Zhi-Hua Zhou. In addition, six tutorials by Włodzisław Duch, Chun Che Fung, Irwin King, Saed Sayad, Jun Tani and M. Emin Yuksel were part of ICONIP 2009. Also, for the first time, there was a Post-ICONIP Workshop held in a neighboring country to the host: the Workshop on Advances in Intelligent Computing (WAIC 2009) was held in Kuala Lumpur, Malaysia on December 7, 2009. Furthermore, the Third International Conference on Advances in Information Technology (IAIT2009) was collocated with ICONIP 2009.

We are indebted to the members of the conference Advisory Board as well as the Governing Board and Past Presidents of APNNA for their advice and assistance in the organization and promotion of ICONIP 2009. We are thankful to the Program Committee and Technical Committee members for their dedication and support in providing rigorous and timely reviews, especially for the last round of submissions due to our extended submission deadline. Each paper was reviewed by at least two referees and three or more reviews were provided in most of the cases. The Program Committee Chairs opted to use the relatively new OCS system and we put it through a rigorous workout and helped the system to smooth out numerous minor issues. We sincerely apologize for any inconvenience the authors may have experienced during the entire paper submission and reviewing process.

A special thanks to the Conference Secretariat, Olarn Rojanapornpun, who worked tirelessly to facilitate many of the conference delegates and to produce these final proceedings. The Organizing Committee members would like to express our sincere appreciation to the devoted behind-the-scene work by Wannida Soontreerutana, Chompoonut Watcharinkorn, Paweena Mongkolpongsiri, Thanyapat Natwaratit, Chutikarn Hongpitakkul, Korakot Eadjongdee, Suda Kasikitsakunphon, Kanittha Charoensuk and Monthana Hunjinda. Last but not least, the organizers gratefully acknowledge the contribution and support from all speakers, panelists and authors, as well as all other participants, in making ICONIP 2009 a resounding success.

December 2009

Jonathan H. Chan
Chi Sing Leung
Minho Lee

Organization

Organizer

School of Information Technology (SIT), King Mongkut's University of Technology Thonburi (KMUTT), Bangkok, Thailand.

Sponsor

Asia Pacific Neural Network Assembly (APNNA)

Technical Co-sponsors

International Neural Network Society (INNS)

Japanese Neural Network Society (JNNS)

European Neural Network Society (ENNS)

IEEE Computational Intelligence Society (IEEE CIS)

Conference Committee

Honorary Chair Shun-ichi Amari, Japan

Advisory Board

Irwin King, Hong Kong	Tom Gedeon, Australia
Nikola Kasabov, New Zealand	Takeshi Yamakawa, Japan
Soo-Young Lee, Korea	Włodzisław Duch, Poland
Derong Liu, USA	Lipo Wang, Singapore
Nikhil R. Pal, India	Jun Wang, Hong Kong
Shiro Usui, Japan	

Local Steering Committee	Borworn Papasratorn, Thailand
	Wichian Chutimaskul, Thailand
	Chakarida Nulkoolkit, Thailand

General Chair	Jonathan H. Chan, Thailand
---------------	----------------------------

Program Chair and Co-chair	Andrew Chi-Sing Leung, Hong Kong
	Minho Lee, Korea

Local Organizing Chair	Kittichai Lavangnananda, Thailand
------------------------	-----------------------------------

VIII Organization

Special Sessions Chairs	Masumi Ishikawa, Japan Tom Gedeon, Australia Shaoning Pang, New Zealand
Workshops Chairs	Lai Weng Kin, Malaysia Chee-Peng Lim, Malaysia
Tutorials Chair	Kevin Wong, Australia
Competitions Chair	John Sum, Taiwan
Publicity Chair	Suree Funilkul, Thailand
Publication Chair	Kriengkrai Porkaew, Thailand
Posters and Demonstration Chair	Vithida Chongsuphajaisiddhi, Thailand
Local Arrangements Chair	Vajirasak Vanijja, Thailand
Conference Secretariat	Olarn Rojanapornpun, Thailand

Program Committee

Shigeo Abe, Japan	Sabri Arik, Turkey
Siu Yeung Cho, Singapore	Yoonsuck Choe, USA
Doo-Hyun Choi, Korea	Seungjin Choi, Korea
Tommy W.S. Chow, Hong Kong	Fu Lai Chung, Hong Kong
Andrzej Cichocki, Japan	Kenji Doya, Japan
Ke-Lin Du, Canada	Tom Gedeon, Australia
Kaizhu Huang, China	Masumi Ishikawa, Japan
Daijin Kim, Korea	Sungshin Kim, Korea
Seong Kong, USA	Takio Kurita, Japan
James Tin-Yau Kwok, Hong Kong	Seong-Whan Lee, Korea
Soo-Young Lee, Korea	Yun-Jung Lee, Korea
Frank Leung, Hong Kong	Chunshien Li, Taiwan
Bao-Liang Lu, China	Bob McKay, Korea
Takashi Morie, Japan	Takashi Omori, Japan
Toshiaki Omori, Japan	Seiichi Ozawa, Japan
Nikhil R. Pal, India	Shaoning Pang, New Zealand
Hyeyoung Park, Korea	Jagath C. Rajapakse, Singapore
Naoyuki Sato, Japan	John Sum, Taiwan
Dianhui Wang, Australia	Young-Gul Won, Korea
Hau San Wong, Hong Kong	Zenglin Xu, Germany
Haixuan Yang, China	Zhirong Yang, Finland
Byoung-Ju Yun, Korea	Zhigang Zeng, China
Kun Zhang, Finland	Liming Zhang, China

Technical Committee

Shotaro Akaho	Tetsuya Asai	Hideki Asoh
Sang-Woo Ban	Hong Bao	Cesar Caiafa
B Chandra	Atthawut Chanthaphan	Shen Chen
Songcan Chen	Xi Chen	Seong-Pyo Cheon
Eng Yeow Cheu	Andrew Chiou	Heeyoul Choi
Ho-Hyoung Choi	Ji Ryang Chung	Zhaohong Deng
Yongtae Do	Bin Dong	Justin Dauwels
Hiroshi Dozono	Béla A. Frigyik	Minoru Fukumi
GH Gan	Eikou Gonda	Raj Gupta
Rodolfo Haber	Hisashi Handa	Osman Hassab Elgawi
Ken Hawick	Hanlin He	Xingshi He
Zhashui He	Akinori Hidaka	Hee-Dong Hong
Jin-Hyuk Hong	Antti Honkela	Xiaolin Hu
Xuelei Hu	Wei Huang	Yung-Fa Huang
Hiroaki Inayoshi	Hiroataka Inoue	Masato Inoue
Rolly Intan	Takaichi Ito	Tae-Seok Jin
Martin Johnson	Norhaslinda Kamaruddin	Keisuke Kameyama
Dae-Seong Kang	Hyohyeong Kang	Satoru Kato
Fujimura Kikuo	Jung-Gu Kim	Kye-Hyeon Kim
Kwang-Baek Kim	Sung hin Kim	Taesu Kim
Yong-Deok Kim	Yong-Tae Kim	YoungIk Kim
Yun-Ho Ko	Takanori Koga	Markus Koskela
Ryosuke Kubota	Jaerock Kwon	Johnny Lai
Choon Young Lee	Daewon Lee	Hyung-Soo Lee
Kwanyong Lee	Nung Kion Lee	Sang-Woong Lee
Gang Li	Gary C. L. Li	Xi Li
Xiangming Li	Xiao-Dong Li	Yufeng Li
Yen-Lun Liang	Chee Peng Lim	Dudy Lim
Heejin Lim	Kil-Taek Lim	Naiyan Lima
Iuon-Chang Lin	Wilfred Lin	Qingshan Liu
Rujie Liu	Weixiang Liu	Yu Liu
Zhiyong Liu	Danniel Cavalcante Lopes	Jacek Mańdziuk
Timothy Mann	Ohkita Masaaki	Seiji Miyoshi
Kenji Nagata	Mitsuteru Nakamura	Wakako Nakamura
Hidehiro Nakano	Yok Yen Nguwi	Qiang Ni
Kenji Nishida	Ikuko Nishikawa	Tohru Nitta
Richard Jayadi Oentaryo	Tetsuya Onoda	Matashige Oyabu
Tomoko Ozeki	Han-Saem Park	Hyung-Min Park
Kiyong Park	Lae-Jeong Park	Sehoon Park
Seongbae Park	Sunho Park	Ekachai Phaisangittisagul
Anh Huy Phan	Kriengkrai Porkaew	Santitham Prom-on
Masukur Rahman	Piyush Rai	Myung-Cheol Roh
Hosaka Ryosuke	Ryo Saegusa	Stefan Schliebs
Gourab Sen Gupta	Daming Shi	Hiroyuki Shimai
Mats Sjöberg	Kingkarn Sookhanaphibarn	Shiliang Sun

Javan Tan	Dacheng Tao	Takashi Takahashi
Masahiro Takatsuka	Mieko Tanaka-Yamawaki	Teik Toe Teoh
Chan Wai Ting	Meng-Hsiun Tsai	Whye Loon Tung
Hiroaki Wagatsuma	Hiroshi Wakuya	Liang Wan
Bo-Hyeun Wang	Jun Wang	Rongjie Wang
Zhanshan Wang	Zhongsheng Wang	Kazuho Watanabe
Bunthit Watanapa	Ginny Wong	Arthur Liang-chuan Wu
Jing Wu	Jiunn-lin Wu	Wei Wu
Zhenping Xie	Hao Xiong	Lu Xu
Yang Xu	Tomoyuki Yamaguchi	Hee-Deok Yang
Huei-Fang Yang	Shengji (Sophie) Yao	Xucheng Yin
Noha Yousri	Yingwei Yu	Zhiwen Yu
Jingling Yuan	Shiu Yin Yuen	Jeong-Min Yun
Rafal Zdunek	Haijun Zhang	He Zhang
Shaohong Zhan	ZhanCheng Zhang	Lei Zheng

Special Session Organizers

Intelligent Data Mining	Kevin Wong
Data Mining for Cybersecurity	Tao Ban Daisuke Inoue Shaoning Pang Youki Kadobayashi
Towards Brain-inspired Systems	Keiichi Horio
SOM and Related Subjects and its Applications	Nobuo Matsuda Heizo Tokutaka Masahiro Takatsuka
Neural Networks for Data Mining	Furao Shen Zhi-Hua Zhou
Hybrid and Adaptive Systems for Computer Vision and Robot Control	Napoleon Reyes Andre Barczak Pitoya Hartono
Artificial Spiking Neural Systems: Nonlinear Dynamics and Engineering Applications	Toshimichi Saito Hiroyuki Torikai
Computational Advances in Bioinformatics	Asawin Meechai Santitham Prom-on
Evolutionary Neural Networks: Theory and Practice	Sung-Bae Cho Kyung-Joong Kim

Local Sponsors

IEEE Thailand Section

Thailand Chapter of ACM

Software Park Thailand

Electrical Engineering/Electronics, Computer, Telecommunications and Information
Technology Association of Thailand (ECTI)

National Electronics and Computer Technology Center (NECTEC)

Table of Contents – Part I

Cognitive Science and Computational Neuroscience

Hebbian-Based Neural Networks for Bottom-Up Visual Attention Systems	1
<i>Ying Yu, Bin Wang, and Liming Zhang</i>	
Modeling of Cortical Signals Using Optimized Echo State Networks with Leaky Integrator Neurons	10
<i>Hanying Zhou, Yongji Wang, and Jiangshuai Huang</i>	
Comparison of Near-Threshold Characteristics of Flash Suppression and Forward Masking	19
<i>Kenji Aoki, Hiroki Takahashi, Hideaki Itoh, and Kiyohiko Nakamura</i>	
Some Computational Predictions on the Possibilities of Three-Dimensional Properties of Grid Cells in Entorhinal Cortex.....	26
<i>Tanvir Islam and Yoko Yamaguchi</i>	
Data Modelling for Analysis of Adaptive Changes in Fly Photoreceptors	34
<i>Uwe Friederich, Daniel Coca, Stephen Billings, and Mikko Juusola</i>	
A Computational Model of Spatial Imagery Based on Object-Centered Scene Representation	49
<i>Naoyuki Sato</i>	
Biophysical Modeling of a <i>Drosophila</i> Photoreceptor	57
<i>Zhuoyi Song, Daniel Coca, Stephen Billings, Marten Postma, Roger C. Hardie, and Mikko Juusola</i>	
Comparing a Cognitive and a Neural Model for Relative Trust Dynamics	72
<i>S. Waqar Jaffry and Jan Treur</i>	
A Next Generation Modeling Environment PLATO: Platform for Collaborative Brain System Modeling.....	84
<i>Shiro Usui, Keiichiro Inagaki, Takayuki Kannon, Yoshimi Kamiyama, Shunji Satoh, Nilton L. Kamiji, Yutaka Hirata, Akito Ishihara, and Hayaru Shouno</i>	

Neurodynamics

Modeling Geomagnetospheric Disturbances with Sequential Bayesian Recurrent Neural Networks	91
<i>Lahcen Ouarbya and Derrick T. Mirikitani</i>	
Finding MAPs Using High Order Recurrent Networks	100
<i>Emad A.M. Andrews and Anthony J. Bonner</i>	
A Study on Bayesian Learning of One-Dimensional Linear Dynamical Systems	110
<i>Takuto Naito and Keisuke Yamazaki</i>	
Decoding Characteristics of D/A Converters Based on Spiking Neurons	118
<i>Masao Takiguchi and Toshimichi Saito</i>	
Separable Recursive Training Algorithms with Switching Module	126
<i>Vijanth S. Asirvadam</i>	
Application-Driven Parameter Tuning Methodology for Dynamic Neural Field Equations	135
<i>Lucian Alecu and Hervé Frezza-Buet</i>	
Interspike Interval Statistics Obtained from Non-homogeneous Gamma Spike Generator	143
<i>Kantaro Fujiwara, Kazuyuki Aihara, and Hideyuki Suzuki</i>	

Mathematical Modeling and Analysis

A Novel Method for Progressive Multiple Sequence Alignment Based on Lempel-Ziv	151
<i>Guoli Ji, Congting Ye, Zijiang Yang, and Zhenya Guo</i>	
Variational Bayes from the Primitive Initial Point for Gaussian Mixture Estimation	159
<i>Yuta Ishikawa, Ichiro Takeuchi, and Ryohei Nakano</i>	
A Bayesian Graph Clustering Approach Using the Prior Based on Degree Distribution	167
<i>Naoyuki Harada, Yuta Ishikawa, Ichiro Takeuchi, and Ryohei Nakano</i>	
Common Neighborhood Sub-graph Density as a Similarity Measure for Community Detection	175
<i>Yoonseop Kang and Seungjin Choi</i>	
Divergence, Optimization and Geometry	185
<i>Shun-ichi Amari</i>	

Robust Stability of Fuzzy Cohen-Grossberg Neural Networks with Delays	194
<i>Tingwen Huang and Zhigang Zeng</i>	
An Adaptive Threshold in Joint Approximate Diagonalization by the Information Criterion	204
<i>Yoshitatsu Matsuda and Kazunori Yamaguchi</i>	
PPoSOM: A Multidimensional Data Visualization Using Probabilistic Assignment Based on Polar SOM	212
<i>Yang Xu, Lu Xu, Tommy W.S. Chow, and Anthony S.S. Fong</i>	
Slice Oriented Tensor Decomposition of EEG Data for Feature Extraction in Space, Frequency and Time Domains	221
<i>Qibin Zhao, Cesar F. Caiafa, Andrzej Cichocki, Liqing Zhang, and Anh Huy Phan</i>	
Stereo Map Surface Calculus Optimization Using Radial Basis Functions Neural Network Interpolation	229
<i>Allan David Garcia de Araujo, Adriaio Duarte Doria Neto, and Allan de Medeiros Martins</i>	
Quasi-Deterministic Partially Observable Markov Decision Processes	237
<i>Camille Besse and Brahim Chaib-draa</i>	
Hierarchical Text Classification Incremental Learning	247
<i>Shengli Song, Xiaofei Qiao, and Ping Chen</i>	
Robust Stability of Stochastic Neural Networks with Interval Discrete and Distributed Delays	259
<i>Song Zhu, Yi Shen, and Guici Chen</i>	
Hybrid Hopfield Architecture for Solving Nonlinear Programming Problems	267
<i>Fabiana Cristina Bertoni and Ivan Nunes da Silva</i>	
Fault Tolerant Regularizers for Multilayer Feedforward Networks	277
<i>Deng-yu Qiao, Chi Sing Leung, and Pui Fai Sum</i>	
Integrating Simulated Annealing and Delta Technique for Constructing Optimal Prediction Intervals	285
<i>Abbas Khosravi, Saeid Nahavandi, and Doug Creighton</i>	
Robust Local Tangent Space Alignment	293
<i>Yubin Zhan and Jianping Yin</i>	
Probabilistic Combination of Multiple Evidence	302
<i>Heeyoul Choi, Anup Katake, Seungjin Choi, Yoonseop Kang, and Yoonsuck Choe</i>	

FIA: Frequent Itemsets Mining Based on Approximate Counting in Data Streams 312
Younghee Kim, Joonsuk Ryu, and Ungmo Kim

Advances in PARAFAC Using Parallel Block Decomposition 323
Anh Huy Phan and Andrzej Cichocki

An Observation Angle Dependent Nonstationary Covariance Function for Gaussian Process Regression 331
Arman Melkumyan and Eric Nettleton

Kernel and Related Methods

DOA Estimation of Multiple Convolutively Mixed Sources Based on Principle Component Analysis 340
Weidong Jiao, Shixi Yang, and Yongping Chang

Weighted Data Normalization Based on Eigenvalues for Artificial Neural Network Classification 349
Qingjiu Zhang and Shiliang Sun

The Optimization of Kernel CMAC Based on BYY Learning 357
Guoqing Liu, Suiping Zhou, and Daming Shi

Closest Source Selection Using IVA and Characteristic of Mixing Channel 365
Choong Hwan Choi, Jae-Kwon Yoo, and Soo-Young Lee

Decomposition Mixed Pixels of Remote Sensing Image Based on 2-DWT and Kernel ICA 373
Huaiying Xia and Ping Guo

Echo Energy Estimation in Active Sonar Using Fast Independent Component Analysis 381
Dongmin Jeong, Kweon Son, Yonggon Lee, and Minho Lee

Improving SVM Classification with Imbalance Data Set 389
Zhi-Qiang Zeng and Ji Gao

Framework for Object Tracking with Support Vector Machines, Structural Tensor and the Mean Shift Method 399
Bogusław Cyganek

Suitable ICA Algorithm for Extracting Saccade-Related EEG Signals . . . 409
Arao Funase, Motoaki Mouri, Andrzej Cichocki, and Ichi Takumi

Learning Algorithms

Learning of Mahalanobis Discriminant Functions by a Neural Network	417
<i>Yoshifusa Ito, Hiroyuki Izumi, and Cidambi Srinivasan</i>	
Implementing Learning on the SpiNNaker Universal Neural Chip Multiprocessor	425
<i>Xin Jin, Alexander Rast, Francesco Galluppi, Mukaram Khan, and Steve Furber</i>	
Learning Gaussian Process Models from Uncertain Data	433
<i>Patrick Dallaire, Camille Besse, and Brahim Chaib-draa</i>	
A Bootstrap Artificial Neural Network Based Heterogeneous Panel Unit Root Test in Case of Cross Sectional Independence	441
<i>Christian de Peretti, Carole Siani, and Mario Cerrato</i>	
A Novel Hierarchical Constructive BackPropagation with Memory for Teaching a Robot the Names of Things	451
<i>Fady Alnajjar, Abdul Rahman Hafiz, and Kazuyuki Murase</i>	
Cellular Neural Networks Template Training System Using Iterative Annealing Optimization Technique on ACE16k Chip	460
<i>Selcuk Sevgen, Eylem Yucel, and Sabri Arik</i>	
Estimation of Driving Phase by Modeling Brake Pressure Signals	468
<i>Hiroki Mima, Kazushi Ikeda, Tomohiro Shibata, Naoki Fukaya, Kentaro Hitomi, and Takashi Bando</i>	
Optimal Hyperparameters for Generalized Learning and Knowledge Discovery in Variational Bayes	476
<i>Daisuke Kaji and Sumio Watanabe</i>	
Backpropagation Learning Algorithm for Multilayer Phasor Neural Networks	484
<i>Gouhei Tanaka and Kazuyuki Aihara</i>	
SNIWD: Simultaneous Weight Noise Injection with Weight Decay for MLP Training	494
<i>John Sum and Kevin Ho</i>	
Tracking in Reinforcement Learning	502
<i>Matthieu Geist, Olivier Pietquin, and Gabriel Fricout</i>	
Ensembling Heterogeneous Learning Models with Boosting	512
<i>Diego S.C. Nascimento and André L.V. Coelho</i>	
Improvement Algorithm for Approximate Incremental Learning	520
<i>Tadahiro Oyama, H. Kipsang Choge, Stephen Karungaru, Satoru Tsuge, Yasue Mitsukura, and Minoru Fukumi</i>	

A Meta-learning Method Based on Temporal Difference Error	530
<i>Kunikazu Kobayashi, Hiroyuki Mizoue, Takashi Kuremoto, and Masanao Obayashi</i>	
Local Learning Rules for Nonnegative Tucker Decomposition	538
<i>Anh Huy Phan and Andrzej Cichocki</i>	
Comparing Large Datasets Structures through Unsupervised Learning	546
<i>Guénaél Cabanes and Younès Bennani</i>	
Applying Duo Output Neural Networks to Solve Single Output Regression Problem	554
<i>Pawalai Kraipeerapun, Somkid Amornsamankul, Chun Che Fung, and Sathit Nakkrasae</i>	
An Incremental Learning Algorithm for Resource Allocating Networks Based on Local Linear Regression	562
<i>Seiichi Ozawa and Keisuke Okamoto</i>	
Learning Cooperative Behaviours in Multiagent Reinforcement Learning	570
<i>Somnuk Phon-Amnuaisuk</i>	
Generating Tonal Counterpoint Using Reinforcement Learning	580
<i>Somnuk Phon-Amnuaisuk</i>	
Robust Approximation in Decomposed Reinforcement Learning	590
<i>Takeshi Mori and Shin Ishii</i>	
Learning of Go Board State Evaluation Function by Artificial Neural Network	598
<i>Hiroki Tomizawa, Shin-ichi Maeda, and Shin Ishii</i>	
Quick Maximum Power Point Tracking of Photovoltaic Using Online Learning Neural Network	606
<i>Yasushi Kohata, Koichiro Yamauchi, and Masahito Kurihara</i>	
Pattern Analysis	
Semi-Naïve Bayesian Method for Network Intrusion Detection System	614
<i>Mrutyunjaya Panda and Manas Ranjan Patra</i>	
Speaker Recognition Using Pole Distribution of Speech Signals Obtained by Bagging CAN2	622
<i>Shuichi Kurogi, Seitaro Sato, and Kota Ichimaru</i>	

Fast Intra Mode Decision for H.264/AVC Based on Directional Information of I4MB	630
<i>Kyung-Hee Lee, En-Jong Cha, and Jae-Won Suh</i>	
Palmprint Recognition Based on Local DCT Feature Extraction	639
<i>H. Kipsang Choge, Tadahiro Oyama, Stephen Karungaru, Satoru Tsuge, and Minoru Fukumi</i>	
Representative and Discriminant Feature Extraction Based on NMF for Emotion Recognition in Speech	649
<i>Dami Kim, Soo-Young Lee, and Shun-ichi Amari</i>	
Improvement of the Neural Network Trees through Fine-Tuning of the Threshold of Each Internal Node	657
<i>Hirotoomo Hayashi and Qiangfu Zhao</i>	
A Synthesis Method of Gene Networks Having Cyclic Expression Pattern Sequences by Network Learning	667
<i>Yoshihiro Mori and Yasuaki Kuroe</i>	
Face Analysis and Processing	
Gender Identification from Thai Speech Signal Using a Neural Network	676
<i>Rong Phoophuangpairaj, Sukanya Phongsuphap, and Supachai Tangwongsan</i>	
Gender Classification Based on Support Vector Machine with Automatic Confidence	685
<i>Zheng Ji and Bao-Liang Lu</i>	
Multiple Occluded Face Detection Based on Binocular Saliency Map ...	693
<i>Bumhwi Kim, Sang-Woo Ban, and Minho Lee</i>	
A Mutual Information Based Face Recognition Method	701
<i>Iman Makaremi and Majid Ahamdi</i>	
Basis Selection for 2DLDA-Based Face Recognition Using Fisher Score	708
<i>Peratham Wiriyathamabhum and Boonserm Kijsirikul</i>	
A Robust Keypoints Matching Strategy for SIFT: An Application to Face Recognition	716
<i>Minkook Cho and Hyeyoung Park</i>	
Selecting, Optimizing and Fusing ‘Salient’ Gabor Features for Facial Expression Recognition	724
<i>Ligang Zhang and Dian Tjondronegoro</i>	

Self-Organized Gabor Features for Pose Invariant Face Recognition	733
<i>Saleh Aly, Naoyuki Tsuruta, and Rin-ichiro Taniguchi</i>	
Image Processing	
Image Hierarchical Segmentation Based on a GHSOM	743
<i>Esteban José Palomo, Enrique Domínguez, Rafael Marcos Luque, and José Muñoz</i>	
An Efficient Coding Model for Image Representation	751
<i>Zhiqing Li, Zhiping Shi, Zhixin Li, and Zhongzhi Shi</i>	
SSTEM Cell Image Segmentation Based on Top-Down Selective Attention Model	759
<i>Sangbok Choi, Sang Kyoo Paik, Yong Chul Bae, and Minho Lee</i>	
Data Partitioning Technique for Online and Incremental Visual SLAM	769
<i>Nopparit Tongprasit, Aram Kawewong, and Osamu Hasegawa</i>	
Improvement of Image Modeling with Affinity Propagation Algorithm for Semantic Image Annotation	778
<i>Dong Yang and Ping Guo</i>	
An Image Identifier Based on Hausdorff Shape Trace Transform	788
<i>Rerkchai Fooprateepsiri, Werasak Kurutach, and Sutthipong Tamsumpaolerd</i>	
Personalized Fingerprint Segmentation	798
<i>Xinjian Guo, Yilong Yin, and Zhichen Shi</i>	
Automatic Image Restoration Based on Tensor Voting	810
<i>Toan Nguyen, Jonghyun Park, Soohyung Kim, Hyukro Park, and Gueesang Lee</i>	
Robust Incremental Subspace Learning for Object Tracking	819
<i>Gang Yu, Zhiwei Hu, and Hongtao Lu</i>	
Reversible Data Hiding Using the Histogram Modification of Block Image	829
<i>Hyang-Mi Yoo, Sang-Kwang Lee, Young-Ho Suh, and Jae-Won Suh</i>	
A Rock Structure Recognition System Using FMI Images	838
<i>Xu-Cheng Yin, Qian Liu, Hong-Wei Hao, Zhi-Bin Wang, and Kaizhu Huang</i>	

Financial Applications

Analyzing Price Data to Determine Positive and Negative Product Associations	846
<i>Ayhan Demiriz, Ahmet Cihan, and Ufuk Kula</i>	
Production Planning Algorithm and Software for Sofa Factory	856
<i>Cholticha Sangngam and Chantana Phongpensri (Chantrapornchai)</i>	
Ensemble Learning for Imbalanced E-commerce Transaction Anomaly Classification	866
<i>Haiqin Yang and Irwin King</i>	
Exploring Early Classification Strategies of Streaming Data with Delayed Attributes	875
<i>Mónica Millán-Giraldo, J. Salvador Sánchez, and V. Javier Traver</i>	
Exchange Rate Forecasting Using Classifier Ensemble	884
<i>Zhi-Bin Wang, Hong-Wei Hao, Xu-Cheng Yin, Qian Liu, and Kaizhu Huang</i>	

Erratum

Backpropagation Learning Algorithm for Multilayer Phasor Neural Networks	E1
<i>Gouhei Tanaka and Kazuyuki Aihara</i>	
Author Index	893

Table of Contents – Part II

Computer Vision

Obstacle Categorization Based on Hybridizing Global and Local Features	1
<i>Jeong-Woo Woo, Young-Chul Lim, and Minho Lee</i>	
Defect Detection and Classification in Citrus Using Computer Vision	11
<i>Jose J. Lopez, Emanuel Aguilera, and Maximo Cobos</i>	
Superresolution from Occluded Scenes	19
<i>Wataru Fukuda, Atsunori Kanemura, Shin-ichi Maeda, and Shin Ishii</i>	
Generating Self-organized Saliency Map Based on Color and Motion	28
<i>Satoru Morita</i>	
Co-occurrence of Intensity and Gradient Features for Object Detection	38
<i>Akinori Hidaka and Takio Kurita</i>	

Control and Robotics

Adaptive Sensor-Driven Neural Control for Learning in Walking Machines	47
<i>Poramate Manoonpong and Florentin Wörgötter</i>	
A Method to Switch Multiple CAN2s for Variable Initial Temperature in Temperature Control of RCA Cleaning Solutions	56
<i>Shuichi Kurogi, Hiroshi Yuno, and Yohei Koshiyama</i>	
Vision-Motor Abstraction toward Robot Cognition	65
<i>Fady Alnajjar, Abdul Rahman Hafiz, Indra Bin Mohd. Zin, and Kazuyuki Murase</i>	
Adaptively Coordinating Heterogeneous Robot Teams through Asynchronous Situated Coevolution	75
<i>Abraham Prieto, Francisco Bellas, and Richard J. Duro</i>	
RL-Based Memory Controller for Scalable Autonomous Systems	83
<i>Osman Hassab Elgawi</i>	

A Semantic SLAM Model for Autonomous Mobile Robots Using
Content Based Image Retrieval Techniques 93
Choon Ling Tan, Simon Egerton, and Velappa Ganapathy

Evolutionary Computation

Parameter Estimation Using a SCE Strategy 107
Pengfei Li, Hesheng Tang, and Zhaoliang Wang

A Novel Evolving Clustering Algorithm with Polynomial Regression for
Chaotic Time-Series Prediction 114
*Harya Widiputra, Henry Kho, Lukas, Russel Pears, and
Nikola Kasabov*

A Multi-strategy Differential Evolution Algorithm for Financial
Prediction with Single Multiplicative Neuron 122
Chukiat Worasuchep and Prabhas Chongstitvatana

Boosted Neural Networks in Evolutionary Computation..... 131
Martin Holeňa, David Linke, and Norbert Steinfeldt

Improving Prediction Interval Quality: A Genetic Algorithm-Based
Method Applied to Neural Networks 141
Abbas Khosravi, Saeid Nahavandi, and Doug Creighton

Involving New Local Search in Hybrid Genetic Algorithm for Feature
Selection 150
Md. Monirul Kabir, Md. Shahjahan, and Kazuyuki Murase

Pareto Optimal Based Evolutionary Approach for Solving
Multi-Objective Facility Layout Problem..... 159
*Kazi Shah Nawaz Ripon, Kyrre Glette, Omid Mirmotahari,
Mats Høvin, and Jim Tørresen*

Other Emerging Computational Methods

Swarm Reinforcement Learning Algorithm Based on Particle Swarm
Optimization Whose Personal Bests Have Lifespans 169
Hitoshi Iima and Yasuaki Kuroe

Effectiveness of Intrinsically Motivated Adaptive Agent for Sustainable
Human-Agent Interaction 179
Takayuki Nozawa and Toshiyuki Kondo

RAST: A Related Abstract Search Tool..... 189
Shiro Usui, Nilton L. Kamiji, Tatsuki Taniguchi, and Naonori Ueda

An Artificial Bee Colony Algorithm for the Quadratic Knapsack Problem	196
<i>Srikanth Pulikanti and Alok Singh</i>	
Universal Learning Machines	206
<i>Włodzisław Duch and Tomasz Maszczyk</i>	
Swarm Diversity Based Text Summarization	216
<i>Mohammed Salem Binwahlan, Naomie Salim, and Ladda Suanmali</i>	
A Fuzzy Bi-level Pricing Model and a PSO Based Algorithm in Supply Chains	226
<i>Ya Gao, Guangquan Zhang, Jie Lu, and Hui-Ming Wee</i>	
Growing Particle Swarm Optimizers with a Population-Dependent Parameter	234
<i>Chihiro Kurosu, Toshimichi Saito, and Kenya Jin'no</i>	
An Efficient Feature Selection Using Ant Colony Optimization Algorithm	242
<i>Md. Monirul Kabir, Md. Shahjahan, and Kazuyuki Murase</i>	
Stable Training Method for Echo State Networks Running in Closed-Loop Based on Particle Swarm Optimization Algorithm	253
<i>Qingsong Song, Zuren Feng, and Yonggang Wang</i>	
Signal, Data and Text Processing	
A Concept Generation Method Based on Mutual Information Quantity among Multiple Self-organizing Maps	263
<i>Kunio Kitahara and Akira Hirose</i>	
Decoding Ambisonic Signals to Irregular Loudspeaker Configuration Based on Artificial Neural Networks	273
<i>Peter Wai-Ming Tsang, Wai Keung Cheung, and Chi Sing Leung</i>	
Document Clustering with Cluster Refinement and Non-negative Matrix Factorization	281
<i>Sun Park, Dong Un An, ByungRea Char, and Chul-Won Kim</i>	
Hierarchical Multi-view Fisher Discriminant Analysis	289
<i>Qiaona Chen and Shiliang Sun</i>	
Auditory Temporal Assimilation: A Discriminant Analysis of Electrophysiological Evidence	299
<i>Hiroshige Takeichi, Takako Mitsudo, Yoshitaka Nakajima, Gerard B. Remijn, Yoshinobu Goto, and Shozo Tobimatsu</i>	

Web Snippet Clustering Based on Text Enrichment with Concept Hierarchy	309
<i>Supakpong Jinarat, Choochart Haruechaiyasak, and Arnon Rungsawang</i>	
Maintaining Footprint-Based Retrieval for Case Deletion	318
<i>Ning Lu, Jie Lu, and Guangquan Zhang</i>	
Investigation of Neonatal EEG Time Series Using a Modified Nonlinear Dynamical Analysis	326
<i>Suparerk Janjarasjitt, Mark S. Scher, and Kenneth A. Loparo</i>	
Solving Fuzzy Linear Regression with Hybrid Optimization	336
<i>M.H. Mashinchi, M.A. Orgun, and M. Mashinchi</i>	
Automatic Document Tagging in Social Semantic Digital Library	344
<i>Xiaomei Xu and Zhendong Niu</i>	
Text Mining with an Augmented Version of the Bisecting K-Means Algorithm	352
<i>Yutaro Hatagami and Toshihiko Matsuka</i>	
Ontology Based Personalized Modeling for Type 2 Diabetes Risk Analysis: An Integrated Approach	360
<i>Anju Verma, Maurizio Fiasché, Maria Cuzzola, Pasquale Iacopino, Francesco C. Morabito, and Nikola Kasabov</i>	
Artificial Spiking Neural Systems: Nonlinear Dynamics and Engineering Applications	
A Pulse-Coupled Network of SOM	367
<i>Kai Kinoshita and Hiroyuki Torikai</i>	
A Simple Spiking Neuron with Periodic Input: Basic Bifurcation and Encoding Function	376
<i>Shimon Teshima and Toshimichi Saito</i>	
Exploiting Temporal Noises and Device Fluctuations in Enhancing Fidelity of Pulse-Density Modulator Consisting of Single-Electron Neural Circuits	384
<i>Andrew Kilinga Kikombo, Tetsuya Asai, and Yoshihito Amemiya</i>	
Bifurcation Analysis of a Resonate-and-Fire-Type Digital Spiking Neuron	392
<i>Tetsuya Hishiki and Hiroyuki Torikai</i>	
Strange Responses to Fluctuating Inputs in the Hindmarsh-Rose Neurons	401
<i>Ryosuke Hosaka, Yutaka Sakai, and Kazuyuki Aihara</i>	

Towards Brain-Inspired Systems

Evaluation of Color Constancy Vision Algorithm for Mobile Robots	409
<i>Yasunori Takemura and Kazuo Ishii</i>	
Surprise-Driven Exploration with Rao-Blackwellized Particle Filters for Efficiently Constructing Occupancy Grid Maps	420
<i>Youbo Cai and Masumi Ishikawa</i>	
Retrieving Emotion from Motion Analysis: In a Real Time Parallel Framework for Robots	430
<i>Tino Lourens and Emilia Barakova</i>	
Using Biologically Inspired Visual Features and Mixture of Experts for Face/Nonface Recognition	439
<i>Zeinab Farhoudi and Reza Ebrahimpour</i>	
Diagnosis Support System for Mucous Membrane Diseases in Oral Cavity	449
<i>Keiichi Horio, Shuhei Matsumoto, Taishi Ohtani, Manabu Habu, Kazuhiro Tominaga, and Takeshi Yamakawa</i>	
Using Long and Short Term Memories in Solving Optimization Problems	457
<i>Masahiro Nagamatu and Jagath Weerasinghe</i>	

Computational Advances in Bioinformatics

Overlap-Based Similarity Metrics for Motif Search in DNA Sequences	465
<i>Hai Thanh Do and Dianhui Wang</i>	
An Evolutionary Artificial Neural Network for Medical Pattern Classification	475
<i>Shing Chiang Tan, Chee Peng Lim, Kay Sin Tan, and Jose C. Navarro</i>	
Coevolutionary Method for Gene Selection and Parameter Optimization in Microarray Data Analysis	483
<i>Yingjie Hu and Nikola Kasabov</i>	
An Omnibus Permutation Test on Ensembles of Two-Locus Analyses for the Detection of Purely Epistatic Multi-locus Interactions	493
<i>Waranyu Wongseree, Anunchai Assawamakin, Theera Piroonratana, Saravudh Sinsomros, Chanin Limwongse, and Nachol Chaiyaratana</i>	
Protein Fold Prediction Problem Using Ensemble of Classifiers	503
<i>Abdollah Dehzangi, Somnuk Phon Amnuaisuk, Keng Hoong Ng, and Ehsan Mohandesi</i>	

Combination of Multiple Features in Support Vector Machine with Principal Component Analysis in Application for Alzheimer’s Disease Diagnosis 512
Jiann-Der Lee, Shau-Chiuan Su, Chung-Hsien Huang, J.J. Wang, Wen-Chuin Xu, You-You Wei, and S.T. Lee

Data Mining for Cybersecurity

Hierarchical Core Vector Machines for Network Intrusion Detection 520
Ye Chen, Shaoning Pang, Nikola Kasabov, Tao Ban, and Youki Kadobayashi

String Kernel Based SVM for Internet Security Implementation 530
Zbynek Michlovský, Shaoning Pang, Nikola Kasabov, Tao Ban, and Youki Kadobayashi

Automated Log Analysis of Infected Windows OS Using Mechanized Reasoning 540
Ruo Ando

HumanBoost: Utilization of Users’ Past Trust Decision for Identifying Fraudulent Websites 548
Daisuke Miyamoto, Hiroaki Hazeyama, and Youki Kadobayashi

A Methodology for Analyzing Overall Flow of Spam-Based Attacks 556
Jungsuk Song, Daisuke Inoue, Masashi Eto, Mio Suzuki, Satoshi Hayashi, and Koji Nakao

A Proposal of Malware Distinction Method Based on Scan Patterns Using Spectrum Analysis 565
Masashi Eto, Kotaro Sonoda, Daisuke Inoue, Katsunari Yoshioka, and Koji Nakao

Evolutionary Neural Networks: Theory and Practice

A Transductive Neuro-Fuzzy Force Control: An Ethernet-Based Application to a Drilling Process 573
Agustin Gajate, Rodolfo Haber, and Pastora Vega

Sentiment Classification with Support Vector Machines and Multiple Kernel Functions 583
Tanasanee Phienthrakul, Boonserm Kijirikul, Hiroya Takamura, and Manabu Okumura

Improving the Performance of Fuzzy ARTMAP with Hybrid Evolutionary Programming: An Experimental Study 593
Shing Chiang Tan and Chee Peng Lim

“Dead” Chromosomes and Their Elimination in the Neuro-Genetic Stock Index Prediction System	601
<i>Jacek Mańdziuk and Marcin Jaruszewicz</i>	
String Pattern Recognition Using Evolving Spiking Neural Networks and Quantum Inspired Particle Swarm Optimization	611
<i>Haza Nuzly Abdull Hamed, Nikola Kasabov, Zbynek Michlovský, and Siti Mariyam Shamsuddin</i>	
Fault Condition Recognition Based on PSO and KPCA	620
<i>Hongxia Pan, Xiuye Wei, and Xin Xu</i>	
Evaluation of Distance Measures for Speciated Evolutionary Neural Networks in Pattern Classification Problems	630
<i>Kyung-Joong Kim and Sung-Bae Cho</i>	
Emergence of Different Mating Strategies in Artificial Embodied Evolution	638
<i>Stefan Elfving, Eiji Uchibe, and Kenji Doya</i>	
Hybrid and Adaptive Systems for Computer Vision and Robot Control	
A Markov Model for Multiagent Patrolling in Continuous Time	648
<i>Jean-Samuel Marier, Camille Besse, and Brahim Chaib-draa</i>	
Hybrid Framework to Image Segmentation	657
<i>Fernando C. Monteiro</i>	
On the Robustness of Fuzzy-Genetic Colour Contrast Fusion with Variable Colour Depth	667
<i>Heesang Shin, Alwyn Husselmann, and Napoleon H. Reyes</i>	
Navel Orange Blemish Identification for Quality Grading System	675
<i>MingHui Liu, Gadi Ben-Tal, Napoleon H. Reyes, and Andre L.C. Barczak</i>	
A Cyclostationarity Analysis Applied to Scaled Images	683
<i>Babak Mahdian and Stanislav Saic</i>	
Intelligent Data Mining	
Non-segmented Document Clustering Using Self-Organizing Map and Frequent Max Substring Technique	691
<i>Todsanai Chumwatana, Kok Wai Wong, and Hong Xie</i>	
A Visual Method for High-Dimensional Data Cluster Exploration	699
<i>Ke-Bing Zhang, Mao Lin Huang, Mehmet A. Orgun, and Quang Vinh Nguyen</i>	

An Algorithm Based on the Construction of Braun’s Cathode Ray Tube as a Novel Technique for Data Classification 710
Mariusz Swiecicki

Fuzzy Decision Tree Induction Approach for Mining Fuzzy Association Rules 720
Rolly Intan and Oviliani Yenty Yuliana

AdaIndex: An Adaptive Index Structure for Fast Similarity Search in Metric Spaces 729
Tao Ban, Shanqing Guo, Qiuliang Xu, and Youki Kadobayashi

Neural Networks for Data Mining

The Application of Wavelet Neural Network Optimized by Particle Swarm in Localization of Acoustic Emission Source 738
Aidong Deng, Li Zhao, and Xin Wei

Speaker Recognition Based on GMM with an Embedded TDNN 746
Cunbao Chen and Li Zhao

Finding Appropriate Turning Point for Text Sentiment Polarity 754
Haipeng Wang, Lin Shang, Xinyu Dai, and Cunyan Yin

Research on Natural Disaster Risk Assessment Model Based on Support Vector Machine and Its Application 762
Junfei Chen, Shihao Zhao, Weihao Liao, and Yuan Weng

Identifying Tobacco Control Policy Drivers: A Neural Network Approach 770
Xiaojiang Ding, Susan Bedingfield, Chung-Hsing Yeh, Ron Borland, David Young, Sonja Petrovic-Lazarevic, and Ken Coghill

Intrusion Detection Using Neural Networks: A Grid Computing Based Data Mining Approach 777
Marcello Castellano, Giuseppe Mastronardi, and Gianfranco Tarricone

SOM and Related Subjects and Its Applications

Recurrent Neural Networks as Local Models for Time Series Prediction 786
Aymen Cherif, Hubert Cardot, and Romuald Boné

Construction of the General Physical Condition Judgments System Using Acceleration Plethysmogram Analysis Results 794
Heizo Tokutaka, Eikou Gonda, Yoshio Maniwa, Masashi Yamamoto, Toshiyuki Kakihara, Masahumi Kurata, Kikuo Fujimura, Li Shigang, and Masaaki Ohkita

Decision of Class Borders on Spherical SOM and Its Visualization	802
<i>Nobuo Matsuda, Heizo Tokutaka, and Matashige Oyabu</i>	
Quantifying the Path Preservation of SOM-Based Information Landscapes	812
<i>Michael Bui and Masahiro Takatsuka</i>	
Self-Organizing Neural Grove and Its Parallel and Distributed Performance	820
<i>Hiroataka Inoue</i>	
The Finding of Weak-Ties by Applying Spherical SOM and Association Rules	828
<i>Takaichi Ito and Tetsuya Onoda</i>	
Analysis of Robustness of Pareto Learning SOM to Variances of Input Vectors	836
<i>Hiroshi Dozono and Masanori Nakakuni</i>	
Interactive Hierarchical SOM for Image Retrieval Visualization	845
<i>Yi Liu and Masahiro Takatsuka</i>	
Evaluation Patterns of Japanese Representative Athletes in the 2008 Beijing Olympic Games: Visualization of Social Expectation and Satisfaction by Use of Self-Organizing Maps	855
<i>Tetsuya Onoda</i>	
Temporal Signal Processing by Feedback SOM: An Application to On-line Character Recognition Task	865
<i>Hiroshi Wakuya and Akira Terada</i>	
A Study on Clustering Method by Self-Organizing Map and Information Criteria	874
<i>Satoru Kato, Tadashi Horiuchi, and Yoshio Itoh</i>	
Author Index	883

Hebbian-Based Neural Networks for Bottom-Up Visual Attention Systems

Ying Yu¹, Bin Wang^{1,2}, and Liming Zhang¹

¹ Department of Electronic Engineering, Fudan University, Shanghai 200433, P.R. China

² The Key Lab of Wave Scattering and Remote Sensing Information (Ministry of Education),
Fudan University, Shanghai 200433, P.R. China
yuying.mail@163.com, {wangbin, lmzhang}@fudan.edu.cn

Abstract. This paper proposes a bottom-up attention model based on pulsed Hebbian-based neural networks that simulate the lateral surround inhibition of neurons with similar visual features. The visual saliency can be represented in binary codes that simulate neuronal pulses in the human brain. Moreover, the model can be extended to the pulsed cosine transform that is very simple in computation. Finally, a dynamic Markov model is proposed to produce the human-like stochastic attention selection. Due to its good performance in eye fixation prediction and low computational complexity, our model can be used in real-time systems such as robot navigation and virtual human system.

Keywords: Visual attention, Bottom-up, Saliency, Pulsed cosine transform, Principal component analysis, Hebbian learning rule.

1 Introduction

In human visual system, there exists a bottom-up attention selection mechanism that can make our eyes rapidly gaze towards salient objects in a clustered scene without any top-down guidance. It is believed that bottom-up visual attention acts like a “spot-light” that can rapidly shift across the entire visual field and selects a small area from the entire visual scene. Only the attended part of input sensory information is allowed to reach short-term memory and visual awareness. So, instead of fully processing the massive sensory input in parallel, a serial mechanism has evolved because of resource limitations [1].

Itti et al. [2] proposed a biologically plausible model of bottom-up attention selection. After that, Walther [3] extended this model to attend to proto object regions and created the Saliency Toolbox (STB). Since Itti and Koch’s model has very complex network architecture, it suffers from computational complexity and over-parameterization.

It is well known that Hebbian learning rule commonly exists among neurons in the human brain [4]. So, Hebbian-based neural networks have been deeply investigated. However, the relationship between Hebbian-based neural networks and selective visual attention has seldom been investigated. In this paper, we only use simple feed-forward, Hebbian-based neural networks to produce visual saliency. The output of networks is binarized (“flattened”) to simulate the lateral surround inhibition of

neurons with similar visual features. Since the orthonormal weights of Hebbian networks are usually used in principal component analysis (PCA) [5][6][7], such computational model is called pulsed PCA (P²CA) transform in this article. Moreover, since discrete cosine transform (DCT) is closely related to the PCA transform, our PCA-based model can be extended to a DCT-based framework [8][9]. This DCT-based attention model is referred to as the pulsed cosine transform (PCT) in this article. Particularly, the visual saliency in our work can be represented in binary codes that simulate neuronal pulses in the human brain. This kind of encoding of visual saliency largely reduces the dynamic range in the state space.

The saliency map guides where the attentional focus is to be deployed, that is, to the most salient location in the scene. Existing attention models shift the attentional focus over different locations with decreasing saliency. However, human eye fixations are not the result of pure bottom-up attention selection, but the result of a combination of bottom-up and top-down attention selection [1]. Movement of attentional focus across the visual field is known to be stochastic rather than deterministic [13]. In order to mimic the human vision system, a dynamic Markov model (DMM) is proposed to conduct stochastic attention selection. Specifically, the more salient a location is, the more probable it will be attended.

Section 2 gives an overview of the proposed architecture of bottom-up visual attention as well as its biological plausibility. Section 3 introduces a human-like stochastic attention selection based on DMM. Section 4 presents experimental results, where our model is compared with STB. Finally, section 5 concludes the paper.

2 Model Architecture

Principal component analysis (PCA) is a powerful technique that has been widely applied in signal feature extraction and dimensionality reduction [4]. Numerous works have been done on how to compute principal components of input data. It has been noted that Hebbian learning in neural networks can find the principal components of incoming sensory data [5][6][7]. Several efficient numerical methods such the EVD and the QR algorithm [16] can also obtain the principal component vectors.

However, the relationship between the principal components of natural scenes and visual salience has seldom been investigated. In this section, we propose the pulsed PCA transform and the pulsed cosine transform to compute the bottom-up saliency map. We will explain how such frameworks relate to visual salience.

2.1 The P²CA Model

To begin with, we propose our computation model of visual attention. Given the input image M , our PCA-based model to compute the saliency map is as follows:

$$P = \text{sign} (C(M)), \quad (1)$$

$$F = \text{abs} (C^{-1}(P)), \quad (2)$$

$$SM = G * F^2, \quad (3)$$

where C and C^{-1} denote the PCA transform and its inverse transform, respectively. $\text{sign}(\cdot)$ is the signum function, and $\text{abs}(\cdot)$ is the absolute value function. G is a 2-dimensional Gaussian low-pass filter.

Equation (1) is called pulsed PCA (P^2CA) transform since it only retains the sign of principal components and discards the amplitude information. Its binary codes (-1 and 1) simulate the neuronal pulses in the human brain. The network architecture of equation (1) is illustrated in Fig. 1. Feedforward connections of PCA transform are trained by Hebbian learning rule. Then, the saliency map is computed by equation (2) and (3). Note that a given image is sub-sampled before its computation. The size of sub-sampling determines the attention scale. In this paper, the input image is resized to be 64×64 pixels.

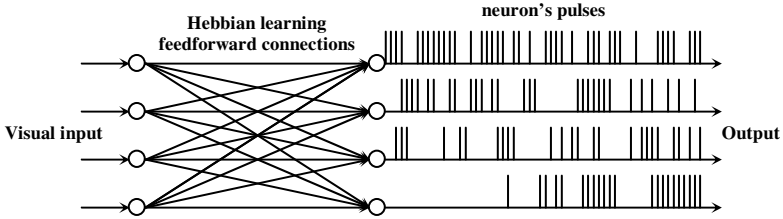


Fig. 1. The feedforward pulsed neural networks for saliency information

It is believed that primitive visual features such as color, edge and intensity are closely related with visual saliency, and they are processed in parallel at pre-attentive stage [10]. According to this theory, we first compute these feature maps respectively before integrating them as a whole. If r , g , and b are the red, green, and blue values of the color image, then the intensity map is computed as

$$M_I = (r + g + b) / 3. \quad (4)$$

Since only one attention scale is considered, instead of using red-green and blue-yellow center-surround opponencies [3], we compute three color maps of red, green, and blue as follows:

$$M_R = r - (g + b) / 2, \quad (5a)$$

$$M_G = g - (r + b) / 2, \quad (5b)$$

$$M_B = b - (r + g) / 2. \quad (5c)$$

Here, M_R , M_G , and M_B are set to zero at locations with negative value. Such color computation is similar with the broadly-tuned color model proposed by [2]. To avoid large fluctuations of the color values at low luminance and balance all original feature maps, the weight factor of each map is calculated:

$$w_I = \max(M_I), \quad (6a)$$

$$w_R = \max(M_R), \quad (6b)$$

$$w_G = \max(M_G), \quad (6c)$$

$$w_B = \max(M_B). \quad (6d)$$

Then, we have

$$F = w_R F_R + w_G F_G + w_B F_B + w_I F_I, \quad (7)$$

where F_R , F_G , F_B , and F_I are computed by equation (1) and (2) with feature maps M_R , M_G , M_B , and M_I , respectively. The saliency map for a color image is then computed by equation (3).

2.2 From P²CA to PCT

PCA is a data-dependent technique, and its transform is affected by the statistics of learning data set [4][5][6][7]. In practical applications, however, a data-independent model is more convenient and favorable. In this section, we attempt to find such an attention model based on the principle of P²CA.

It has been noted that the PCA basis will probably come to resemble the DCT basis as the population of learning data with stationary statistics tends to infinite [8][9]. So, one can consider the discrete cosine transform as a particular PCA transform. In our attention model, we can replace the PCA transform by a 2-dimensional discrete cosine transform and ultimately produce a data-independent attention model. This model is called pulsed cosine transform (PCT) in this article.

Note that for PCA, the down-sampled 64×64 image is reshaped into a 4096-dimensional vector before it is projected onto the principal axes (i.e., the eigenvectors of natural images). Different from PCA, the 2-dimensional DCT is a separable decomposition in rows and columns [9]. So, its computational space complexity is much lower than the PCA transform.

It has recently been proposed that the primary visual cortex (V1) creates a bottom-up saliency map, with the location of the most active neuron responding to a scene most likely to be selected [11]. This proposal suggests that the computation of saliency is instantiated in the neural dynamics arising from the lateral surround inhibition by activities of nearby neurons with similar features. Accordingly, neuronal activities come to occur typically at locations of pop-out items, highlighting the breakdown of statistical homogeneity in the input. DCT represents the visual input with periodical signals of different frequency and different orientation. So, large coefficients of DCT contain the information of statistical homogeneity. By flattening the magnitude, PCT mimics the lateral suppression among neurons with similar features. Therefore, our model can compute the saliency map of the input image.

In addition, the visual saliency in our framework can be represented in binary codes. Such binary encoding of saliency information not only simulates neuronal pulses in the human brain but also has lower dynamic range in the state space. Comparing with the investigation in V1, the extent to which higher visual areas, such as V2 and beyond, contribute to pre-attentive selection and attentive influences is as yet unclear [11]. We expect that our framework can become a heuristic model of the pre-attentive mechanism in higher visual cortex.

3 Human-Like Attention Selection

The saliency map guides where the attentional spotlight is to be deployed, that is, to the most salient location in the scene. A plausible neural architecture to detect the most salient location is the winner-take-all (WTA) network [12]. A computational strategy called “inhibition of return” (IOR) [12] was also proposed to avoid attend only to the location of maximal saliency. WTA and IOR are computationally very important components of attention since it allows us to rapidly shift the attentional focus over different locations with decreasing saliency.

As a matter of fact, human eye fixations are not the result of pure bottom-up attention selection, but the result of a combination of bottom-up and top-down attention selection [1]. If we do not take into account top-down influences, eye fixations are determined by a bottom-up saliency map. Pure bottom-up attention selection, however, does not exist. Human attention is known to be stochastic rather than deterministic [13]. When humans scan the same picture, their scanpaths are different between trials even if they have no visual search tasks. In this case, top-down influences comprise many factors such as personal mood, experiences, long or short-term memory, psychological and biophysical conditions. Such top-down influences can be described as a random disturbance to bottom-up visual saliency. Hence, we propose a dynamic Markov model (DMM) to mimic the human visual system and conduct the stochastic attention selection.

Given a saliency map, assume that salient areas L_1, L_2, \dots, L_n are arranged in order of decreasing saliency. Let $P_i(t)$ be the probability of attending L_i at step t and $s_i(t)$ be its instantaneous saliency degree. Let s_i be the value of location i in the saliency map generated by an attentional model. Note that $s_i(0) = s_i$. So, $P_i(t)$ can be computed by:

$$P_i(t) = \frac{s_i(t)}{\sum_j s_j(t)}, \quad i = 1, 2, \dots, n. \quad (8)$$

Computationally, IOR implements a short-term memory of previously visited locations and allows the human visual system to focus on a new location. The simplest implementation of IOR consists of triggering transient inhibition in the saliency map at the currently attended location. Assume that location i was attended at step τ , its instantaneous saliency degree $s_i(t)$ can be computed as:

$$s_i(t) = \begin{cases} \left(\frac{t-\tau-1}{n-1}\right)^\lambda s_i & \tau < t < \tau + n \\ s_i & \text{other} \end{cases}. \quad (9)$$

Here, the amnesic parameter $\lambda > 0$. Usually, let $\lambda = 2$.

Fig. 2 illustrates an example for the shift of the attentional spotlight. As can be seen, the DMM for attention selection allows robots to flexibly shift their attention to a less prominent, but important object (i.e., the helicopter in the top right corner). As compared, by shifting the focus of attention with decreasing saliency, the resulting scanpath is deterministic for any given saliency map. So, the stochastic process produced by DMM is more human-like than the conventional approach.

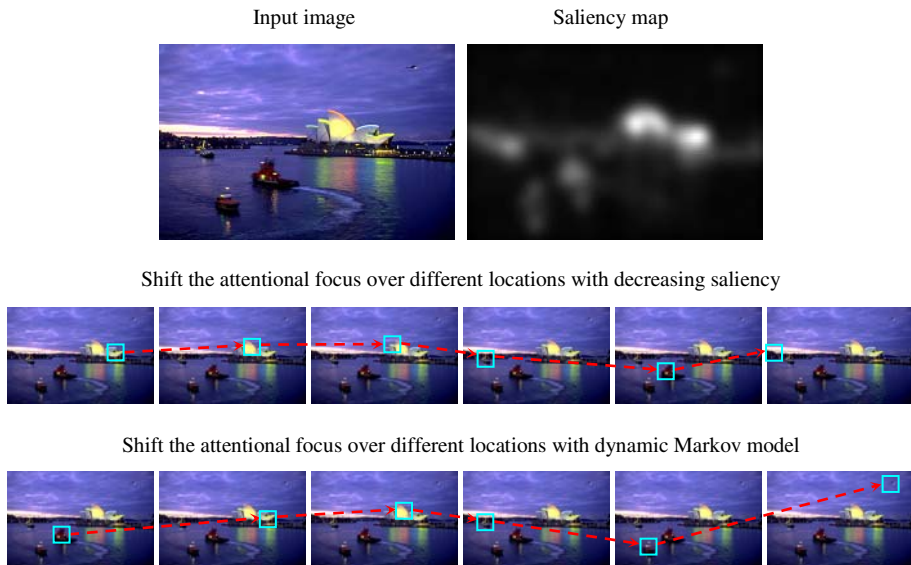


Fig. 2. An example for the shift of the attentional spotlight

4 Experimental Results

This section evaluates the output of the proposed model as compared with Walther’s Saliency Toolbox (STB) [3]. STB is perhaps the most popular model of saliency based attention and currently appears to be the yardstick to measure the performance of other models. We set the saliency maps’ resolution of P²CA and PCT to 64×64 pixels in all experiments. The principal component vectors for P²CA are estimated with one million 64×64 sub-images (image patches) that are gathered by sampling from hundreds of natural images. For STB, we use the default parameters. All experiments in this paper are implemented using Matlab7.0 in such computer environment as Intel 2.53G CPU and 2G Memory.

To measure the consistency of a visual attention model with human eye fixation, we use the database from [14] (containing 120 colored natural images of urban environments) and eye fixation data from 20 subjects as ground truth. We use P²CA, PCT and STB to produce their saliency maps on the 120 images. We use the number and the ratio of correct saliency detection as consistency measure with only the first fixation which is most likely to be driven by bottom-up attention mechanism as proposed in [15]. The results given in Table 1 show that both P²CA and PCT outperform STB in this test.

Table 1. Number and ratio of correct detection in the first fixation

Model	P ² CA	PCT	STB
Number of correct detection	71	73	47
Ratio of correct detection	0.5917	0.6083	0.3917

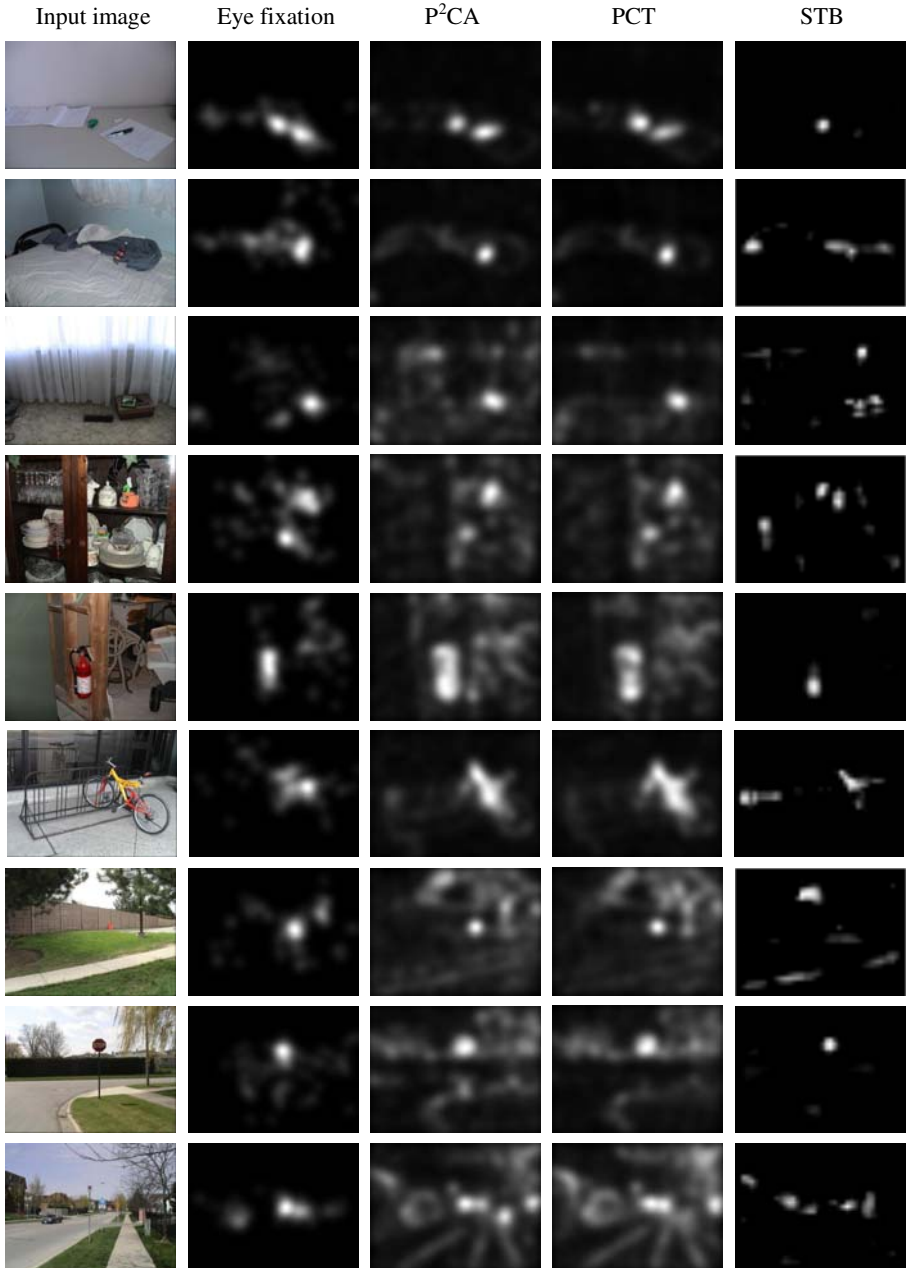


Fig. 3. Some examples of our experimental images

Receiver operating characteristic (ROC) curve is recently used to evaluate the saliency map's ability to predict human fixations in natural images [15]. The larger the ROC area is, the better the prediction power of a saliency map is. So, we calculate the

ROC areas for P²CA, PCT and STB according to all fixations and the first 2 fixations respectively. These results are shown in Table 2. As can be seen, both P²CA and PCT models outperform STB. Note that PCT is marginally better than P²CA.

We also conduct a qualitative comparison of all the models. Fig. 3 illustrates some examples and their eye fixation density maps as ground truth. We can notice a resemblance between P²CA and PCT. That's why their results in Table 1 and Table 2 are similar. Meanwhile, we have compared their computation speed. STB's CPU-time for all 120 natural images is 52.162 seconds. P²CA's CPU-time is 66.111 seconds. PCT takes only 1.488 seconds to compute all saliency maps. So, the PCT model is very fast in computation, which can meet real-time requirements in video systems.

Table 2. ROC areas for different saliency models according to human fixations

Model	P ² CA	PCT	STB
All fixations	0.7796	0.7882	0.6043
First 2 fixations	0.7897	0.7982	0.6183

5 Conclusions

This paper aims to find an attentional model based on Hebbian-based neural networks. The proposed model not only has good performance in eye fixation prediction but also has the biological and developmental implication for the visual attention mechanism. Since our model is very simple and fast in computation, it can be used in engineering field such as robot navigation, virtual human system, and intelligent auto-focus system embedded in digital camera.

Acknowledgments. This work was supported by National Science Foundation of China (Grant No. 60672116) and Shanghai Leading Academic Discipline Project (B112). The authors thank Neil Bruce for kindly sharing the eye fixation data.

References

1. Itti, L., Koch, C.: Computational modeling of visual attention. *Nature Rev. Neurosci.* 2, 194–203 (2001)
2. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Patt. Anal. and Mach. Intell.* 20(11), 1254–1259 (1998)
3. Walther, D., Koch, C.: Modeling attention to salient proto-objects. *Neural Networks* 19, 1395–1407 (2006)
4. Haykin, S.: *Neural Networks - A Comprehensive Foundation*, 2nd edn. Prentice Hall, Englewood Cliffs (2001)
5. Oja, E.: A simplified neuron model as a principal component analyzer. *J. Math. Bio.* 15, 267–273 (1982)
6. Sanger, T.D.: Optimal unsupervised learning in a single-layer linear feedforward neural network. *IEEE Trans. Neural Networks* 2, 459–473 (1989)
7. Weng, J., Zhang, Y., Hwang, W.S.: Candid covariance-free incremental principal component analysis. *IEEE Trans. Patt. Anal. and Mach. Intell.* 25(8), 1034–1040 (2003)

8. Ahmed, N., Natarajan, T., Rao, K.: Discrete cosine transform. *IEEE Trans. Computers*, 23, 90–93 (1974)
9. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*, 2nd edn. Prentice Hall, Englewood Cliffs (2002)
10. Treisman, A.M., Gelade, G.: A feature-integration theory of attention. *Cognitive Psychology* 12(1), 97–136 (1980)
11. Li, Z., Dayan, P.: Pre-attentive visual selection. *Neural Networks* 19, 1437–1439 (2006)
12. Koch, C., Ullman, S.: Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology* 4(4), 219–227 (1985)
13. Brockmann, D., Geisel, T.: The ecology of gaze shifts. *Neurocomputing* 32-33, 643–650 (2000)
14. Bruce, N.D., Tsotsos, J.K.: Saliency based on information maximization. In: *Proc. NIPS* (2005)
15. Tatler, B.W., Baddeley, R.J., Gilchrist, I.D.: Visual correlates of fixation selection: effects of scale and time. *Vision Research* 45, 643–659 (2005)
16. Golub, G.H., van Loan, C.F.: *Matrix Computation*, 3rd edn. John Hopkins University Press, Baltimore (1996)

Modeling of Cortical Signals Using Optimized Echo State Networks with Leaky Integrator Neurons

Hanying Zhou, Yongji Wang*, and Jiangshuai Huang

Key Lab. for Image Processing and Intelligent Control Education Ministry of China
Department of Control Science and Engineering
Huazhong University of Science and Technology
Wuhan, 430074 China
wangyjch@hust.edu.cn

Abstract. Echo State Networks (ESNs) is a newly developed recurrent neural network model. It has a special echo state property which entitles it to model nonlinear dynamic systems whose outputs are determined by previous inputs and outputs. The ESN approach has so far been worked out almost exclusively using standard sigmoid networks. Here we will consider ESNs constructed by leaky integrator neurons, which incorporate a time constant and the dynamics can be slowed down. Furthermore, we optimized relevant parameters of the network by Particle Swarm Optimization (PSO) in order to get a higher modeling precision. Here the input signals are spikes distilled from the monkey's motor cortex in an experiment and the outputs are the moving trajectories of the wrist of a monkey in the experiment. The results show that this model can well translate the neuronal firing activities into the desired positions.

Keywords: Echo State Networks, Leaky Integrator Neurons, Particle Swarm Optimization.

1 Introduction

Since it was proved that it was possible to predict the hand position of a primate using cortical neuronal firing activity in the widely acclaimed article by Wessberg *et al*[1] scholars have been making efforts on building a channel to send messages to outside equipments from brain cortical signals, which is called Brain-Machine Interface (BMI). Along with the development of the technique of multi-channel neural signals collection technology and computer control science, how to get the useful information of neural activities of the brain cortex, and how to model the signals for detailed behavior are the keys for the BMI system.

For different biology experiments and research purposes, the modeling methods vary. Scholars have made deep research in modeling neural spike trains and moving activities [2] and proposed diversified modeling frameworks. The inputs are typically multidimensional neural recordings collected from relevant regions of a monkey's

* Corresponding author.

brain. In this paper, we will utilize ESN to learn the mappings of motor cortical signals and the move gesture of a primate in the Brain-Machine Interface.

ESN is a kind of recurrent neural network so just like the traditional RNNs, it has dynamic property and short-term memory. However, the special training scheme of ESN, which will be introduced in next section, distinguished it from other RNNs and make up for the shortage caused by traditional learning methods which are based on back-propagation of errors.

The disadvantage of the widely used kind of ESN with standard sigmoid neurons is that no time constant is contained so it is impossible to slow down their dynamics, thus they are just suited for modeling inherently discrete-time systems with a “computational”, “jumpy” flavor. They behave not very satisfactorily in some occasions of slow dynamics. However, the ESN approach is not confined to standard sigmoid networks which have been used almost exclusively so far. The basic idea of this net’s property that will be introduced works with different forms.

In this paper we propose to use leaky integrator neurons as the internal units of this net. This model was first used in an ESN context in Jaeger 2001[3]. It is a continuous-time neuron model which contains a time constant and has individual state dynamics that can be exploited in diversified ways to adapt this network to the temporal characteristics of a learning task, so it can solve the problem just stated. The concrete mechanism of this neuron model will be introduced in section 3.

Several parameters of the whole network need to be adjusted manually in order to make the net work more precisely, which is very inconvenient. Here we optimize them with PSO and well solve this important problem.

Here the spike trains are motor cortical signals derived from the motor cortex of a monkey. The spikes are recorded when the monkey was doing a certain arm flexion and extension movements.

The whole paper is arranged as this: after the instruction, we give the definition and structure of the ESN. Then, we setup the experiment and give the outcome of the simulation and draw some conclusion at last.

2 Echo State Networks: Overview

Echo State Network is first proposed by Jaeger [3], and it shares some similarities with the Liquid State Machine which is proposed by Mass *et al* [4] because they both work after the mechanism of dynamic reservoir. In this section, we will briefly summarize the basic principles of ESN.

2.1 Structure

The typical structure [3] of it is shown in Fig.1, which displays that it is composed of an input layer, an internal layer and an output layer. Here the net is a discrete-time neural network and these three layers have K , N and L nodes respectively. The input units are connected to the “reservoir” of N recurrent networks by a $N \times K$ weight matrix W^{in} . The internal units are interconnected with untrained random weights which are collected in a $N \times N$ matrix W . All units can be connected to output units by the $L \times (K + N + L)$ matrix W^{out} . This implies that connections directly from the input

to the output nodes and connections between output nodes are allowed. When ESN is used for some models, there should be feedback connections from output layer to the reservoir and these weights are in the $N \times L$ matrix W^{back} .

The notable character of ESN is that its internal layer is composed of a large number of neurons and the neurons are sparsely connected to each other. This layer is the so-called ‘‘Dynamic Reservoir’’ (DR) and it can map inputs into high-dimensional space and reserve information of the past which is useful.

2.2 Echo State Property

Under certain conditions [5], the network state vector $x(n) = (x_1(n) \dots x_N(n))^T$ is uniquely determined by the left-infinite input history $u(n), u(n-1), \dots$ presented to the network. More precisely, if there exists function series $E = (e_1, \dots, e_N)$ (here $e_i : U^{-N} \rightarrow R$) such that for $\dots, u(n-1), u(n) \in U^{-N}$ the current network state can be expressed by $x(n) = E(\dots, u(n-1), u(n))$, then we say this net has the echo state property and the series of functions are called echo functions.

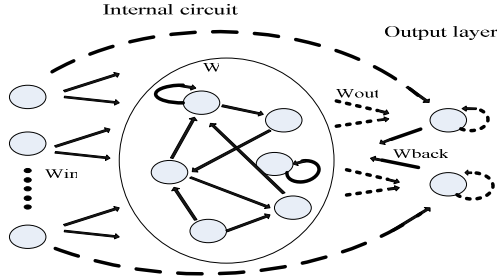


Fig. 1. Block diagram of an Echo State Network (the dash arrows represent the matrix that need to be trained while the solid ones represent those that are fixed)

In [11], the conditions of this property are deeply researched and the conclusions indicated that for ESN with normal sigmoid units, it is sufficient to ensure this property by scaling the spectral radius of internal weight matrix to $|\lambda_{max}| < 1$. The condition for leaky integrator neurons will be introduced in next section.

2.3 Training Procedure

Here we briefly describe the steps of the off-line training procedure [6]:

1. Give the temporal input/output series $(u(n), d(n)), n = 1, 2, 3 \dots T$
2. Generate the values of matrixes (W^{in}, W, W^{back}) randomly and scale W to make the echo state property satisfied. Typically for net with normal sigmoid internal units this can be done by making $|\lambda_{max}| \leq 1$, where λ_{max} is the maximum eigenvalue of W .
3. Drive the net with the training signals given in step 1 by presenting them into the net with W_{in} and W_{back} respectively and thus get the internal states vector $x(n)$. The method varies as for different internal neurons.

4. At each time step, put the network state vector $x(n)$ as a new row into the state collecting matrix M , and at each time collect the inverse of sigmoid teacher outputs $\tanh^{-1} d(n)$ into the teacher collection matrix T in a similar way.
5. The desired weight matrix can be computed by

$$W^{out} = (M^+T)^t \quad (1)$$

Here M^+ represent the pseudo inverse of M and t indicates transpose operation

For exploitation, Feed new input sequences into the trained network and it could predict new outputs.

The calculation of internal states $x(n)$ depends on neural model of internal units and in next section the update equation of it will be given. We calculate outputs by

$$y(n+1) = f^{out}(W^{out}(u(n+1), x(n+1), y(n))) \quad (2)$$

3 ESN with Leaky Integrator Neuron

The evolution of a continuous-time leaky integrator neuron [7] is described by the differential equation

$$\dot{x} = C(-ax + f(W^{in}u + Wx + W^{back}y)) \quad (3)$$

where C is a time constant—the larger, the faster the resulting dynamics, and a is the leaking decay rate. By modeling a decaying potential of the neuron, $-ax$ helps the neuron reserve part of its history state. The larger the decay rate, the faster the attenuate of history state, and the greater the relative affect of input from other neurons. Transform this differential equation into a difference equation with step size μ ($0 < \mu < 1$), we can obtain

$$x(n+1) = (1 - \mu Ca)x(n) + \mu Cf(W^{in}u(n+1) + Wx(n) + W^{back}y(n)) \quad (4)$$

The function $f(\cdot)$ is the standard sigmoid function, $f(\cdot) = \tanh(\cdot)$. Iterate step by step according to this update equation, it can map the previous inputs and outputs to the internal states. Now we will give the conditions under which an ESN with this kind of neurons can work properly. It means that it has the “echo state property” which is discussed in the previous section. Not very rigidly stated, the Echo State Property means that the current states are uniquely decided by the history input values and also the previous outputs if there are feedbacks. It has been proved that scaling the matrix $\mu CW + (1 - \mu Ca)I$ to make the spectral radius of it less than unity can assure existence of echo state^[3]. The internal units should be sparsely interconnected with each other by W . The connection rate is very low in order to make sure that the units can run with enough space and thus obtain affluent dynamics, which is of great importance for the approximation of dynamic systems. The W is randomly generated within proper scope and will be fixed once given.

In the first section, we have introduced the advantage of this kind of neuron. For modeling continuously and slowly transforming systems, using networks with continuous dynamics is obviously more feasible. So in this paper we try to model the motor cortical signals using an ESN which is composed of these leaky integrator neurons.

4 Experiments

The monkey whose motor system in the cerebral cortex was implanted with micro-electrode arrays of multi-channel was trained to study a task named centre-out [8~12]. As represented in Fig.2, the monkey moved a cursor from the starting point to one of eight goals in a 3D imaginary cube. The cursor and goals were shown in the monkey's workspace with outlines, but did not exist physically. When one of the eight lights was on, the monkey reached to it, and the trajectory of the wrist was recorded by the sensor taped to it. The spike trains were noted simultaneously from the neurons in the motor cortex of the monkey and here 38 neurons were noted.

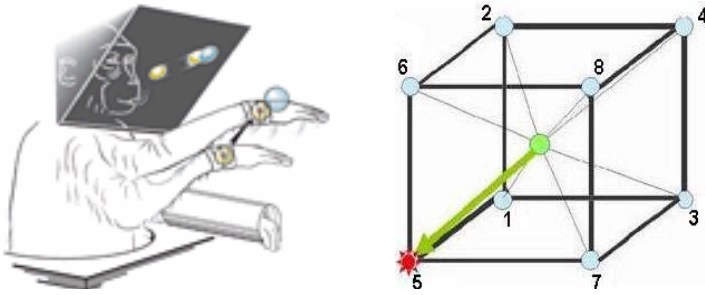


Fig. 2. Sketch map of the experiments and the eight directions we adopt in this experiment

There are eight directions all together and we do 20 experiments repetitively each direction. We obtain 160 sets of data during the monkey's experiment. The trajectories recorded are shown in Fig.3

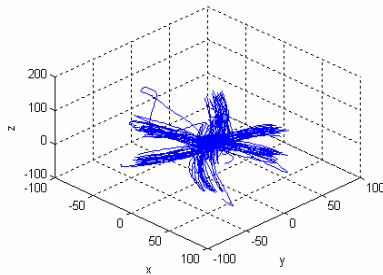


Fig. 3. The recorded trajectories of all the trails

5 Modeling Using ESN

Now we utilize ESN with leaky integrator neurons to design the model for the experiments. First, Find 16 neurons which are closely related with the task using t-test method ($P < 0.05$) and single-factor analysis ($P < 0.05$).

The original spikes recorded can not be directly input into the net so we should set up a time-bin first and count the spike numbers in the bin and this is the so-called ‘‘Frequency code’’ because it is about the frequency of spikes in the bin. Here the neural spike events were binned in windows of 100ms which switch at the step of 20ms.

The structure shown in Fig.1 is actually the most complete form of this net, which is used to deal with auto-regressive systems, whose output $y(n+1)$ depends on previous outputs $y(n)$, $y(n-1)$...as well as history inputs. For the data here, every coordinate of the trajectories is obviously relevant to its previous values, thus this system is a auto-regressive one and it take the structure of Fig.1 for its model.

The number of internal units are chosen to be $N = 500$. The connectivity rate of the recurrent connection matrix W is set to be 1%. The net has 16 input units for 16 neurons and 3 output units which together represent the 3-D coordinates of the moving track.

Furthermore, the weights of the matrix W should be equilibrated in the rough, that is to say, the average value of weights should be about zero [7]. Here we draw nonzero weights randomly over $[-1, 1]$.

The setting of its spectral radius α is of crucial importance for the modeling performance of ESN training because it indicates the speed of teacher dynamics. The absolute size of input weights W^{in} is also of great importance, Large absolute values of W^{in} implies that the network is strongly actuated by inputs while the opposite means that it is only slightly activated around the DR’s zero states. Similar statements hold for the situation of matrix W^{back} . We generate matrixes W^{in} and W^{back} randomly from $[-\lambda_1, \lambda_1]$ and $[-\lambda_2, \lambda_2]$ respectively.

For the leaky integrator neurons, set the parameters $\{a, \mu\}$ to $\{1, 1\}$.

First we try hand-tune those parameters that haven’t been decided, in which way we can just randomly decide their values within a reasonable scope and adjust them according to experience and some principles by simulating again and again. This task is somehow tough and complicated. We tried out more than ten times until we get a satisfactory result with demanded precision, that is, the average distance between the sampled points of the real trajectory and the predicted one is no more than 5cm.

Now we abandon this process and use particle swarm optimization to decide the values of them, which is much more convenient and effective.

The time constant will be optimized with the other three parameters mentioned above.

For every direction, pick up several trails for training and use the rest trails for testing. The 8 different directions are trained and tested respectively and independently.

Suppose for a direction there are j training samples thus there are j teacher time series of $(u_1(n), d_1(n))_{n=0, \dots, n_1}, \dots, (u_j(n), d_j(n))_{n=0, \dots, n_j}$, where $u(n)$ is the 16-dimensional vector of input and $d(n)$ is vector of the corresponding coordinate. As mentioned above, first we should set the parameters $\{\alpha, \lambda_1, \lambda_2, C\}$ using PSO. The

fitness function should reflect represents the total error of all the training steps. To ensure the constraint condition that the spectral radius of $\tilde{W} = \mu CW + (1 - \mu Ca)I$ should satisfy that $|\lambda_{\max}(\tilde{W})| - 1 < 0$, the fitness function is chosen as

$$fitness = \sum_{m=1}^j \sum_{n=0}^{n_j} (d_m(n) - y_m(n))^2 + 10 * \max[0, |\lambda_{\max}(\tilde{W})| - 1] \quad (5)$$

Here each particle is a 4-dimensional vector X_i whose dimensions are for the four parameters. Each particle has a current velocity V_i and a personal best position X_{pbest_i} and we denote the global best position with X_{gbest} . The updating of velocity and values is done according to Eq.6 and Eq.7.

$$V_i(t+1) = wV_i(t) + c_1 r_1 (X_{pbest_i}(t) - X_i(t)) + c_2 r_2 (X_{gbest}(t) - X_i(t)) \quad (6)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (7)$$

where r_1 and r_2 are two random numbers between 0 and 1. c_1 and c_2 are two learning factors and here they are both set to be 2.1 and 2 respectively.

After we get all the proper parameters, we put them for training. Feed each sample fed into the net, update the internal states with

$$x(n+1) = (1 - \mu Ca)x(n) + \mu Cf(W^{in}u(n+1) + Wx(n) + W^{back}d(n)) \quad (8)$$

After this process we can obtain j state matrixes of M_1, M_2, \dots, M_j and teacher matrixes of T_1, T_2, \dots, T_j . Accumulate all these matrixes as this:

$$M = [M_1, M_2, \dots, M_j]^T, T = [T_1, T_2, \dots, T_j]^T \quad (9)$$

Then use Eq.1 to calculate the desired W^{out} and now the whole net is completed and can be used to predict the track by feeding new input signals into it. This process is done according to Eq.4 and Eq.2.

We try to reduce the number of internal processing elements gradually with PSO optimizing the relevant parameters, and by doing this we finally get that we can still get a model with acceptable precision with this number being cut off to 300. The less the internal units, the fast the training process, so this is a very meaningful improvement. The consequences are presented as follows.

In Fig.4, we give a simulating result for a training sample.

In Fig.5, we pick up two of the predict results just for illustration.

From Fig.4, it could be obviously seen that the training precision is very high as the two trajectories almost superpose each other. In Fig.5, the red trajectory, which is a line combined by coordinates computed at each forecast step, can well track the original one.

In this paper, we adopt ESN with leaky integrator neurons to model the cortical signals of the monkey, it could be seen from the test results that this net can get acceptable result in translating the cortical signals into the desired outputs.

The essence of ESN is that it can map the original low-dimensional space to high-dimensional space which is much easier to be read out.

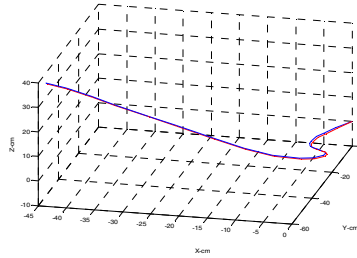


Fig. 4. The simulation result for a training sample (The blue represents the original collected while the red represent the test result)

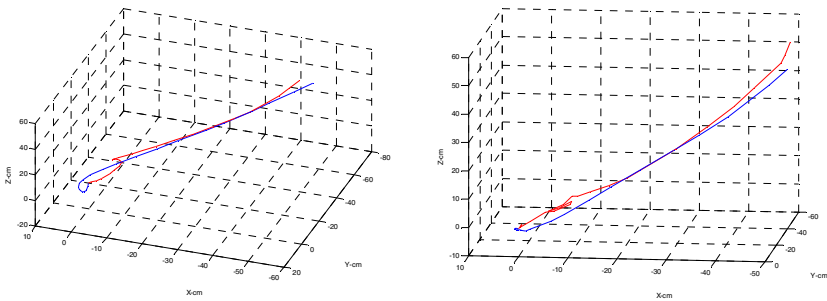


Fig. 5. The testing results of the forecasting track (The blue represents the original collected while the red represent the test results)

Another notable advantage of this net is its easy training procedure. Compared with traditional recurrent network, only the output weight matrix needs to be trained while the other weight matrixes are fixed once given. What's more, the training of it is quite simple because it is computed by a linear regression method. It is quite easy and fast, and obviously this character is of great importance for in-line prediction which demands high speed computing.

In conclusion, we can get satisfactory results using Echo State Networks with leaky integrator neurons.

Acknowledgements

This work is supported in part by National Nature Science Foundation of China under Grant 60674105, 60975058 and High Tech program under Grant 2008AA04Z215, as well as Nature Science Foundation of Hubei Province China under Grant 2007ABA027. We appreciate the data provided by Professor Jiping He who works in the department of bioengineering of Arizona State University, as well as his helpful suggestion in modeling.

References

1. Wessberg, J., Stambaug, C.R., Kralik, J.D., Beck, P.D., Laubach, M., Chapin, J.K., Kin, J., Biggs, S.J., Srinivasan, M.A., Nicolelis, M.A.L.: Real-time Prediction of Hand Trajectory by Ensembles of Cortical Neurons in Primates. *Nature* 408, 361–365 (2000)
2. Schwartz, A.B., Taylor, D.M., Helms Tillery, S.I.: Extraction algorithms for cortical control of arm prosthetics. *Current Opinion in Neurobiology* 11(6), 701–707 (2001)
3. Jaeger, H.: The Echo State Approach to Analyzing and Training Recurrent Neural Networks, GMD Report 148, GMD-German National Research Institute for Computer Science (2001)
4. Maas, W., Natschlager, T., Markram, H.: Real-time Computing without Stable States: A New Framework for Neural Computing Based on Perturbation. *Neural Computing* 14(11), 2531–2560 (2002)
5. Jaeger, H.: Short term memory in echo state networks. *Fraunhofer Institute for Autonomous Intelligent Systems* (2002)
6. Scherer, S., Oubbati, M., Schwenker, F., Palm, G.: Real-Time Emotion Recognition from Speech Using Echo State Networks. In: Prevost, L., Marinai, S., Schwenker, F. (eds.) *ANNPR 2008. LNCS (LNAI)*, vol. 5064, pp. 205–216. Springer, Heidelberg (2008)
7. Jaeger, H.: A tutorial on training recurrent neural networks. *GMD-Report 159*, German National Research Institute for Computer Science (2002)
8. Taylor, D.M., Helms Tillery, S.I., Schwartz, A.B.: Direct cortical control of 3D neuroprosthetic devices. *Science* 296(5574), 1829–1832 (2002)
9. Wahnoun, R., He, J., Helms Tillery, S.I.: Selection and parameterization of cortical neurons for neuroprosthetic control. *Journal of Neural Engineering* 3, 162–171 (2006)
10. Georgopoulos, A.P., Schwartz, A.B., Kettner, R.E.: Neuronal population coding of movement direction. *Science* 233, 1416–1419 (1986)
11. Schwartz, A.B., Kettner, R.E., Georgopoulos, A.P.: Primate motor cortex and free arm movement to visual targets in three-dimensional space. I. Relations between single cell discharge and direction of movement. *J. Neurosci.* 8(8), 2913–2927 (1988)
12. Reina, G.A., Moran, D.W., Schwartz, A.B.: On the relationship between joint angular velocity and motor cortical discharge during reaching. *J. Neurophysiol.* 85, 2576–2589 (2001)

Comparison of Near-Threshold Characteristics of Flash Suppression and Forward Masking

Kenji Aoki, Hiroki Takahashi, Hideaki Itoh, and Kiyohiko Nakamura

Department of Computational Intelligence and Systems Science,
Interdisciplinary Graduate School of Science and Engineering,
Tokyo Institute of Technology,
4259-G3-46 Nagatsuta-cho, Midori-ku, Yokohama, Kanagawa 226-8502, Japan
{aokikenji,taka,hideaki}@brn.dis.titech.ac.jp, nakamura@dis.titech.ac.jp

Abstract. A normally visible stimulus can be rendered invisible by some psychophysical techniques. Flash suppression and forward masking are two such techniques. In this study, we investigated the selectivity of suppression in flash suppression and forward masking. Observers were asked to discriminate the orientation or color of the test grating that was at the orientation-discrimination threshold. We found that during flash suppression color sensitivity was more suppressed than orientation sensitivity. Forward masking produced a pattern of results similar to flash suppression. These results suggest that the flash suppression and forward masking share partly a common mechanism.

Keywords: Flash suppression, forward masking, color, orientation, form.

1 Introduction

Psychophysical techniques that render a normally visible stimulus invisible [1] provide a powerful tool for investigating the neural correlates of conscious visual experience [2,3,4]. One such technique is flash suppression [5]. In flash suppression, a stimulus is first presented to one eye (initial stimulus), followed by presentation of the same stimulus to the same eye (test stimulus) and a dissimilar stimulus to the opposite eye (contralateral stimulus). Under these conditions, observers perceive only the initial and contralateral stimuli, and the test stimulus is rendered invisible [6]. Previous studies showed that the initial and test stimuli need not be spatially similar to obtain the suppressive effect [6,7].

In this study, we focus on the interaction between the initial and test stimuli. The detection threshold of a visual stimulus could be raised by a preceding visual stimulus, without presentation of the contralateral stimulus, which is known as forward masking [8,9]. Hanson and Anderson [10] measured the detection threshold for a monocular color patch after the extinction of the larger stimulus that was presented to the same eye. They reported the observer's inability to identify the color of the detection-threshold color patch. Because flash suppression and forward masking share a common stimulation sequence, i.e., the test stimulus is preceded by the initial stimulus, we predicted that, in flash suppression, observer

may also fail to identify color of the threshold test stimulus, when the initial stimulus is larger than the test stimulus. To test the prediction, we asked observers to discriminate the orientation or color of the test grating presented as the test stimulus in flash suppression. The grating was at the orientation-discrimination threshold. According to the prediction, we anticipated that observer’s performance on the orientation discrimination task would be higher than that on the color discrimination task.

2 Methods

Seven observers participated in the experiment. All of them were naive to the purpose of the experiment. The experiment was conducted in a dark room. Observer’s head movements were reduced using chin and forehead rests. Visual stimuli were presented on a CRT display (SONY CPD-E200). Left half images on the display were presented to the left eye, and right half images on the display were presented to the right eye, through mirror stereoscope. The distance from the display to the eyes was 1 m. All stimuli were presented on a black background.

Each observer’s performances to discriminate the orientation and color of the test stimulus were examined in three stimulation conditions: flash suppression condition, monoptic masking condition, and dichoptic masking condition. The flash suppression condition began with presentation of five small squares to each eye (Fig. 1a). Center squares were fixation points. Observer was instructed to fixate these squares during they were presented. Squares around the center squares were presented to induce binocular fusion. Following the observer’s button press, fixation points disappeared, and a white diamond shape (initial stimulus; side lengths are 3° , 95.3 cd/m^2) was presented to the right eye for 1 s. After a blank field for 40 ms, a test stimulus was presented to the right eye, and a white diamond shape (contralateral stimulus; side lengths are 3° , 95.3 cd/m^2) was presented to the left eye, for 10 ms. Then the stimuli disappeared, and the observer responded using one of two buttons. The test stimulus was a square-wave grating ($1^\circ \times 1^\circ$, 5 c/deg), its orientation was -45° or $+45^\circ$ from vertical, and its color was red or achromatic. The observer’s task was to discriminate the orientation or color of the test grating. In the orientation discrimination tasks, the observer had to press the left (right) button when the -45° ($+45^\circ$) grating was presented. In the color discrimination tasks, the observer had to press the left (right) button when the red (achromatic) grating was presented. After the response, the fixation points reappeared and the next trial followed.

The monoptic and dichoptic masking conditions were the same with the flash suppression condition, except that the contralateral stimulus did not appear in the monoptic and dichoptic condition (Fig. 1b, c) and that the initial stimulus was presented to the left eye in the dichoptic condition (Fig. 1c).

Each block consisted of 120 trials. During one block every pair of the three stimulation conditions and four test stimuli was presented 10 trials in a random order. The observers performed the orientation-discrimination task block and the color-discrimination task block alternately. Each task consisted of 5 blocks.

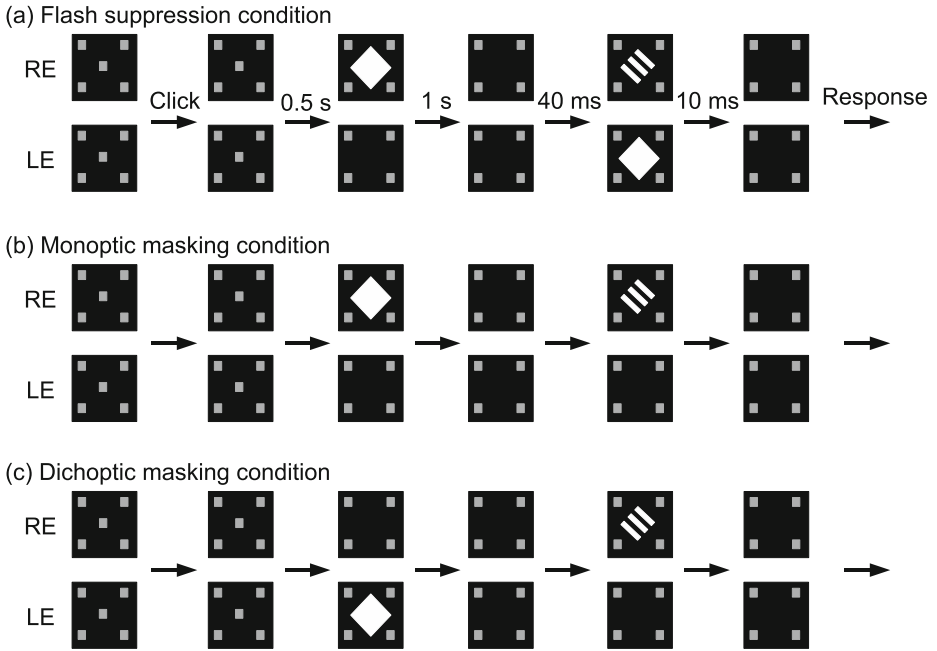


Fig. 1. Time course of the stimulation conditions. The test stimulus was presented in three conditions: (a) flash suppression condition, (b) monoptic masking condition, and (c) dichoptic masking condition. Stimuli, which were presented to the right eye, are shown in RE rows, and stimuli, which were presented to the left eye, are shown in LE rows.

The task order was counterbalanced across the observers. Luminances of the red and achromatic test gratings were at the orientation-discrimination thresholds, which were predetermined using a two-down-one-up staircase strategy for each color in the flash suppression condition [11,12].

3 Results

The averaged percent correct of each condition and task is shown in Fig. 2. In the flash suppression condition, observer's performance on the orientation discrimination task was significantly higher than that on the color discrimination task (Wilcoxon signed rank test, $p < 0.05$). Three of the seven observers showed a significant difference individually ($p < 0.05$; Fig. 3a). In the monoptic masking condition, although no significant difference between performances in the two tasks was shown in a group analysis (Wilcoxon signed rank test, $p = 0.16$), in two of the seven observers, a significant difference was found individually ($p < 0.05$; Fig. 3b). In the dichoptic masking condition, the performance

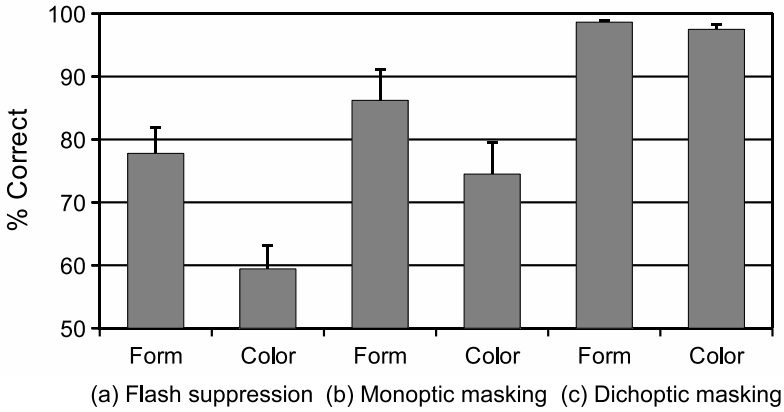


Fig. 2. Averaged performance in each condition and task. (a) Performance in the flash suppression condition. (b) Performance in the monoptic masking condition. (c) Performance in the dichoptic masking condition. The performance was measured by the percent correct. ‘Form’ means the performance of the orientation discrimination tasks, and ‘Color’ means the performance of the color discrimination tasks. Error bars represent standard error of the mean.

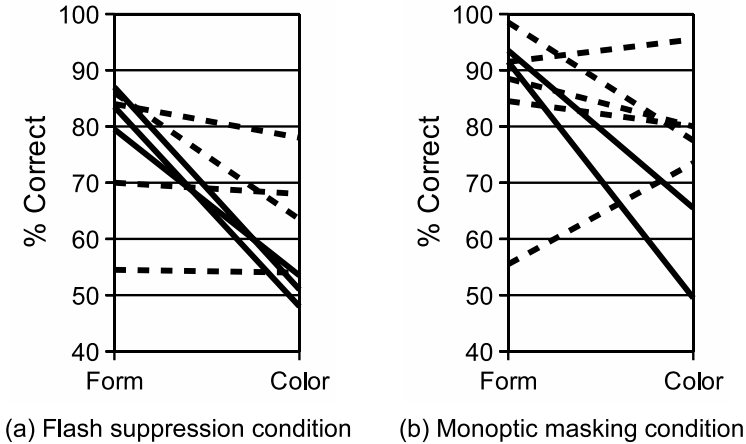


Fig. 3. Individual performance in each condition and task. (a) Performance in the flash suppression condition. (b) Performance in the monoptic masking condition. Performance was measured by the percent correct. Each line represents one observer. Performances of observers who showed a significant difference are shown by solid lines, and performances of the other observers are shown by dashed lines.

of the orientation discrimination task was not significantly higher than the performance of the color discrimination task (sign test, $p = 0.38$), and no observers showed a significant difference individually ($p > 0.7$).

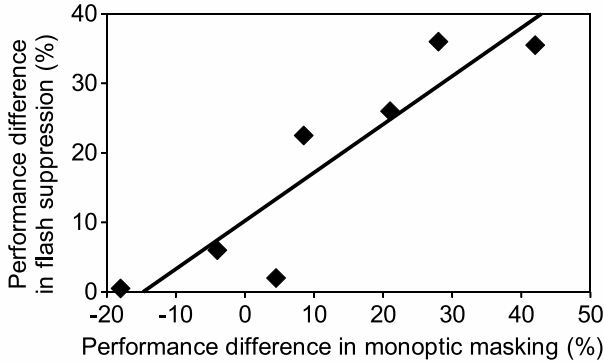


Fig. 4. The performance difference found in the flash suppression condition was significantly correlated with that in the monoptic masking condition. Here, the performance difference is the difference in the percent correct between the orientation discrimination task and the color discrimination task. Each point represents a single subject.

The performance difference between the orientation discrimination task and the color discrimination task in the flash suppression condition was significantly correlated with that in the monoptic masking condition (Spearman’s rank correlation test, $\rho = 0.93$, $p < 0.01$; Fig. 4).

4 Discussion

We found that the orientation or form of the test grating was better discriminated than the color of it in the flash suppression condition and for some observers in the monoptic masking condition. Previous studies also reported loss of color perception of visual stimuli. Hanson and Anderson [10] reported that, following presentation of the monocular masking stimulus, observers failed to identify the color of the detection-threshold test patch. Our results also showed that color was more difficult to discriminate than orientation for two observers in the monoptic masking condition. This suggests that orientation discrimination as well as stimulus detection is easier than color identification.

The flash suppression condition is identical to the monoptic masking condition if the contralateral stimulus is removed. The loss of color discrimination in the monoptic masking suggests a possibility that the inferiority of color discrimination in the flash suppression condition was also caused by the masking effect by the initial stimulus. If so, it is expected that the larger the discrimination performance difference in the monoptic masking condition, the larger the difference in the flash suppression condition. Actually, this was the case (Fig. 4).

Another study reporting loss of color perception was concerned with binocular rivalry. The binocular rivalry is a phenomenon that two dissimilar images presented respectively to two eyes compete for perceptual dominance [13, 14]. Smith et al. [15] measured the detection threshold of the colored test probe that

was presented to the dominant or non-dominant eye. When the threshold test probe was presented to the non-dominant eye, observers failed to perceive its color. This suggests that interaction of signals between the two eyes may also reduce capability of color discrimination. In our results, the dichoptic masking condition did not show inferior color discrimination, suggesting that the contralateral stimulus in the flash suppression condition was not involved in the reduction of color discrimination.

As described above, the present data suggest that the inferior discrimination of color in the flash suppression condition is produced by the monoptic masking effect. However, the dichoptic masking effect still needs to be examined because of the following reason. We determined the orientation discrimination thresholds in the flash suppression condition and used the same threshold test gratings in the dichoptic masking conditions. As a result, performances of the two discrimination tasks were at the ceiling in the dichoptic masking condition (Fig. 2c). There is a possibility that this eliminated possible difference in discrimination performance between orientation and color [16]. Experiments in which the orientation discrimination threshold is respectively determined in each condition need to be conducted.

The flash suppression condition presented here differs from that previously studied in two respects. First, although the initial stimulus was followed by the test stimulus without delay in the previous studies [5,12], they were presented with a delay of 40 ms in the present experiment. Second, while the initial stimulus was the same as the test stimuli in the previous studies, they were different in the present study, just as in the experiment of Hanson and Anderson. Effects of these differences on the discrimination performance are also issues for future research.

Acknowledgments. We thank the reviewers for their valuable comments. This work was partially supported by KAKENHI (19700215).

References

1. Kim, C.Y., Blake, R.: Psychophysical Magic: Rendering the Visible ‘Invisible’. *Trends Cogn. Sci.* 9, 381–388 (2005)
2. Frith, C., Perry, R., Lumer, E.: The Neural Correlates of Conscious Experience: An Experimental Framework. *Trends Cogn. Sci.* 3, 105–114 (1999)
3. Rees, G.: Neuroimaging of Visual Awareness in Patients and Normal Subjects. *Curr. Opin. Neurobiol.* 11, 150–156 (2001)
4. Rees, G., Kreiman, G., Koch, C.: Neural Correlates of Consciousness in Humans. *Nat. Rev. Neurosci.* 3, 261–270 (2002)
5. Ooi, T.L., Loop, M.S.: Visual Suppression and Its Effect upon Color and Luminance Sensitivity. *Vision Res.* 34, 2997–3003 (1994)
6. Wolfe, J.M.: Reversing Ocular Dominance and Suppression in a Single Flash. *Vision Res.* 24, 471–478 (1984)
7. Brascamp, J.W., Knapen, T.H.J., Kanai, R., van Ee, R., van den Berg, A.V.: Flash Suppression and Flash Facilitation in Binocular Rivalry. *J. Vision* 7, 1–12 (2007)

8. Breitmeyer, B.G.: *Visual Masking: An Integrative Approach*. Oxford University Press, New York (1984)
9. Kahneman, D.: Method, Findings, and Theory in Studies of Visual Masking. *Psychol. Bull.* 70, 404–425 (1968)
10. Hanson, J.A., Anderson, E.M.S.: Studies on Dark Adaptation. VII. Effect of Pre-exposure Color on Foveal Dark Adaptation. *J. Opt. Soc. Am.* 50, 965–969 (1960)
11. Levitt, H.: Transformed Up-Down Methods in Psychoacoustics. *J. Acoust. Soc. Am.* 49, 467–477 (1971)
12. Tsuchiya, N., Koch, C., Gilroy, L.A., Blake, R.: Depth of Interocular Suppression Associated with Continuous Flash Suppression, Flash Suppression, and Binocular Rivalry. *J. Vision* 6, 1068–1078 (2006)
13. Leopold, D.A., Logothetis, N.K.: Multistable Phenomena: Changing Views in Perception. *Trends Cogn. Sci.* 3, 254–264 (1999)
14. Blake, R., Logothetis, N.K.: Visual Competition. *Nat. Rev. Neurosci.* 3, 1–11 (2002)
15. Smith, E.L., Levi, D.M., Harwerth, R.S., White, J.M.: Color Vision is Altered During the Suppression Phase of Binocular Rivalry. *Science* 218, 802–804 (1982)
16. Zolman, J.F.: *Biostatistics: Experimental Design and Statistical Inference*. Oxford University Press, New York (1993)

Some Computational Predictions on the Possibilities of Three-Dimensional Properties of Grid Cells in Entorhinal Cortex

Tanvir Islam and Yoko Yamaguchi

Laboratory for Dynamics of Emergent Intelligence, RIKEN BSI,
2-1 Hirosawa, Wako-shi, Saitama, Japan
{tanvir,yokoy}@brain.riken.jp

Abstract. The discovery of grid cells in the entorhinal cortex (EC) of the rat (Hafting et al. 2005) has provided many hints of the mechanism of spatial computation in brain during animal movement. Since then, various experiments as well as computational modeling studies of grid cells have answered some of the key questions related to the properties of these cells. However, almost all of these studies are conducted on the rats and mice during their movement in horizontal space, and it is not clear whether the grid cells possess a three-dimensional firing field during movement in space that is either tilted or curved. In this paper, we make some predictions on the possibilities of three-dimensional shapes of grid fields by hypothesizing that they indeed possess such properties, and produce such three-dimensional fields during movement in tilted space. We show several polyhedral shapes that can be generated by our computational neural network model, and in case of movement in horizontal plane, our three-dimensional grid cell model is reduced to a two-dimensional model to generate grid fields similar to experimental findings.

1 Introduction

Discovery of place cells [1]-[5] in the hippocampal regions of rats consolidated the idea that hippocampus probably represents a cognitive map of the local environment of an animal. Place cells, firing on specific locations of the environment encode the location of the animal, and possess the ability to represent and recall the spatial environment with collective neural representation. However, the source of the input signals for place cells and the underlying mechanism of place fields was still an unsolved problem until Hafting et al. (2005) [6] discovered “grid cells” in EC layer II and III, which give major input to CA3 and CA1. Unlike the place cells, the firing fields of the grid cells create a regularly tessellating grid-like pattern in hexagonal formation, spanned over the horizontal environment where the animal is moving. The strict periodicity of the tessellating firing pattern of grid cells suggested that they are a key element of the spatial navigation system [6] [7]. Grid cells with fields of various spacing, spatial phase, and angular orientations prompted the idea that their ensemble can compute the space, and are majorly responsible for place field formation. Now it is

widely believed that the combination of head direction cells and grid cells probably compute the functionality of path integration in the upstream of hippocampus, collecting pivotal spatial information from sensory system.

Since the discovery of grid cells, several computational models [7][10][11] have been proposed to explain the functional mechanism of the periodic tessellation of the firing patterns. These models are largely of two types: the intracellular oscillator models and the attractor dynamics-based models. We proposed a computational model of grid fields [11] that is based on column structures of HD units and grid cells, expanded over from the EC deep layers towards layer II. In this model, grid fields with various size, spatial phase, and angular orientation are calculated from only velocity and head direction angle inputs to the EC deep layers. However, due to the lack of experimental data of grid cells during movement in environments not horizontal, i.e. tilted or curved space, these models do not assume the possibility three-dimensionality of grid fields. In this paper, we hypothesize that grid fields are originally convex polyhedrons, and we provide a computational neural network model of grid cells in rat entorhinal cortex.

2 Our Hypothesis and Model

Experimental finding and theoretical modeling of grid fields have so far contributed substantially in understanding of the mechanisms behind the computation of space in entorhinal hippocampal network. Observation of the two dimensional hexagonal grid fields generated in rat entorhinal cortex cells raises an obvious question: Do grid fields have three-dimensional properties? What will be the response of the grid cells when an animal will move along a tilted space that constitutes movement in three dimensions? Because the natural movement of rat is restricted to mainly two dimensions, there have not been many experimental studies to find out the three dimensional property of grid cells. Some studies [9] of hippocampal place cells found that during movement of a rat on a tilted track, firing fields of many place cells remap even though the rat moves the same distance on a same track that was initially horizontal. An earlier experiment conducted with rats travelling in a NASA space shuttle [8] showed similar remapping of place cells. In these experiments, the other environmental cue was same in case of movement in both horizontal and tilted track. Because place cells receive their major inputs from EC layer II and III grid cells, these experimental findings now suggest that grid fields themselves may be three dimensional, with showing properties different from horizontal movement to tilted movement, thus causing the remapping of place cells. With not much experimental data published about the three dimensional properties of grid cells, this assumption is confined to the level of prediction at this moment.

We hypothesize that grid fields are actually three dimensional, and the two dimensional grid fields that we observe are special cases of a more generalized property of grid cells. It should be noted that by “three-dimension” we mean space that is not a horizontal plane. Therefore, a tilted conic space (used in the related simulations) or a

slope is three-dimensional in space in our definition because they are not horizontal planes. We assumed that the animal's sense of a 'tilt-angle', which can be relayed from sensory system to EC, is additionally required in our extended model of three-dimensional grid fields. In our 2D model [11], HD cells in the deep layer direction system have preferred directions in 2D space, denoted by a single angle. However, in the case of 3D space, two angles are necessary to express such direction vectors of the system. The first one is the "horizontal angle", which is the angle formed by the 2D projection of a direction vector with the horizontal X-axis, and measured from the X-axis towards the projection of the vector. This angle is basically the same as the direction angles mentioned in the 2D model. The second angle is the "vertical angle", which the direction vector forms with the vertical Z-axis, and is measured from the Z-axis towards the vector. To visualize it, we can think of a three-dimensional polar axis system, where the position of a point in space is defined by one horizontal and one vertical angle. Similarly, instead of a single "head direction angle" input, we can think of also another angle that jointly defines the movement: the "tilt angle" that corresponds to the animal's sense of the slope of a tilted space.

Taking the above into account, the internal calculation of the i th component of the direction system [11] can be expressed as below:

$$\begin{aligned}
S_i^t = & [v_i \{ \sin(\phi_i^{ver}) \cos(\phi_i^{hor}) \sin(\theta_H^{t,ver}) \cos(\theta_H^{t,hor}) \\
& + \sin(\phi_i^{ver}) \sin(\phi_i^{hor}) \sin(\theta_H^{t,ver}) \sin(\theta_H^{t,hor}) \\
& + \cos(\phi_i^{ver}) \cos(\theta_H^{t,ver}) \} \\
& + B \{ \sin(\phi_i^{ver}) \cos(\phi_i^{hor}) \sin(\theta_{B,H}^{t,ver}) \cos(\theta_{B,H}^{t,hor}) \\
& + \sin(\phi_i^{ver}) \sin(\phi_i^{hor}) \sin(\theta_{B,H}^{t,ver}) \sin(\theta_{B,H}^{t,hor}) \\
& + \cos(\phi_i^{ver}) \cos(\theta_{B,H}^{t,ver}) \} + S_i^{t-1}] \bmod A
\end{aligned} \tag{1}$$

Where, at time t , S_i^t is the internal state of the i th component of the direction system, v_i is the velocity input to the system, $\theta_H^{t,hor}$ is the head direction angle input, $\theta_H^{t,ver}$ is the tilt angle input, A is the spacing of grid fields, ϕ_i^{ver} is the angle of direction vector with vertical Z-axis, ϕ_i^{hor} is the angle of direction vector with the horizontal X-axis, $\theta_{B,H}^{t,ver}$ is the vertical angle of spatial phase, $\theta_{B,H}^{t,hor}$ is the horizontal angle of spatial phase, B is the amount of spatial phase.

It can be noted that the amount S_i^t corresponds to the modulus of displacement along the preferred direction of the i th component. The output of the component can be given as:

$$\begin{aligned}
I_i^t = & 1 \quad (\text{if } 0 \leq S_i^t \leq r \text{ or } (A-r) \leq S_i^t \leq A) \\
= & 0 \quad (\text{otherwise})
\end{aligned} \tag{2}$$

Where, r is the radius of the grid field. The output of the grid cell can be given as:

$$D^t = \prod_i I_i^t \quad (3)$$

When the animal is moving in purely horizontal surface, tilt angle input to deep layer is 90 degrees, then, eq. (1) is reduced to:

$$S_i^t = [v_i \sin(\phi_i^{ver}) \cos(\phi_i^{hor} - \theta_H^{t,hor}) + \text{spatial phase term} + S_i^{t-1}] \bmod A \quad (4)$$

For simplicity, in eq. (4) the terms of spatial phase are not shown, but these terms can be assumed accordingly to 3D space. From eq. (4), we can see that the projections of 3D direction vectors to the horizontal 2D surface are similar to the direction vectors mentioned in [11]. The value of $\sin(\phi_i^{ver})$ varies from 0 to 1, which is also responsible for the change in width and spacing of grid fields in 2D environment, besides the parameters A and r .

3 Possible Shapes of 3D Grid Fields

The grid fields observed during 2D horizontal movement are patterned in the form of hexagons [6]. In our 2D model, direction vectors of the hexagonal direction systems have preferred angles along the 2D plane, which create 2D stripes (Fig. 1, left), normal to the vectors. The cross sections of these stripes are the computationally derived grid fields. In case of 3D, direction vectors have preferred angles (2 angles each) along the 3D plane, creating 3D solid disk-like patterns (Fig. 1, right), whose cross sections form the solid shapes of the 3D grid fields we consider.

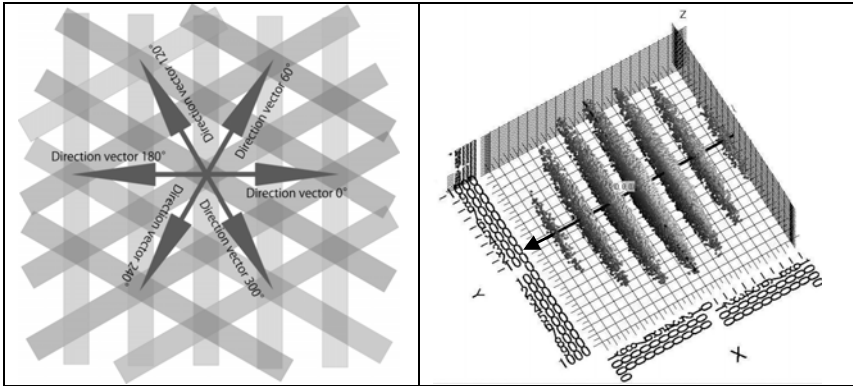


Fig. 1. Example of direction vectors in 2D and 3D model. Left: Stripe-like patterns are formed at normal to the direction of six vectors, with preferred angles 60 degrees apart. The cross sections of these patterns form hexagonal grids. Right: Example of the disk-like fields at normal to a direction vector of vertical angle 90 degrees (with Z-axis) and horizontal angle 0 degree (with X-axis).

The grid fields in our 2D model [11] are hexagon shaped polygons, expanded over the 2D surface in hexagonal formation. Considering the intersections of direction vectors with vertical and horizontal preferred angles, we can assume the 3D grid fields to be polyhedral, with each face of the polyhedron being normal to the preferred direction of the direction vectors whose periodic disk-like fields form the convex polyhedron by intersecting with each other. There is one key constraint for such polyhedral, that is, in case of movement in the horizontal environment, where tilt angle $\theta_H^{t,ver} = 90^\circ$, the resultant grid fields from our 3D model should be similar to hexagonal grid fields observed in experiments. Another constraint might be whether the polyhedron should be space-filling as well, that is, in case of $r = 2r$, the tessellation of the polyhedron covers the whole 3D space without any gap. Though some of the polyhedral we discuss here are space-filling, we consider only the first condition to be the only constraint for our model. For simulation, we used a spherical space with diameter of 900 cm, and also a 30 degrees tilted conic space. To simulate movement of an animal, we used velocity as 10cm/s, grid spacing =300 cm, and grid width=50 cm. The head direction angle and tilt angle are changed randomly in the spherical space.

As shown in Fig. 2, the simplest of the 3D grid field we can think of is the hexagonal column, which is expanded continuously along the vertical Z-axis. The direction vectors in this case are $(\phi_i^{hor}, \phi_i^{ver}) = (0, 90), (60, 90), (120, 90)$ degrees. The projections of the grid fields generated during simulation of movement on a conic space of 30 degrees of tilt angle (Fig. 2) are exactly same as the grid fields generated by the 2D model expressed in eq. (4).

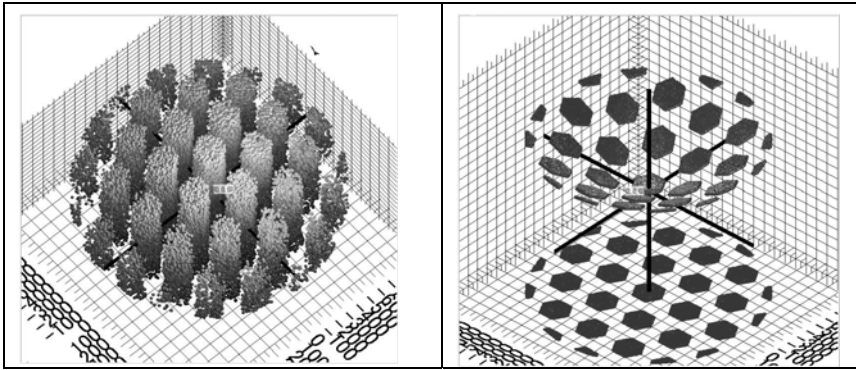


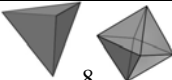
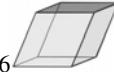


Fig. 2. Left: Simulated hexagonal columns with the 3D grid model. Right: grid fields generated by hexagonal column during movement on a 30 degree tilted conic space. Projections on the XY horizontal plane are shown below.

It can also be assumed that grid fields are discontinuous and aligned in layered formations along the vertical Z-axis. We can think of layered structures of solids with every layer similar (hexagonal prism, hexagonal dipyrmaid etc.), every alternative layers similar (tetrahedron and octahedron couple), or every third layers similar

(Rhombic hexahedron) etc. Among these, the combination of tetrahedron and octahedron is not a single solid, but their combination is space filling. In table 1, we show the direction vectors and other properties of these solids. Furthermore, in Fig. 3 (left column) we show the shapes of these 3D grid fields with simulating movement in 3D space. We also show that in case of all these solids, the 2D grid fields generated by setting tilt angle input as 90 degrees are similar to experimentally found grid fields (Fig. 3 middle column). Again, the firing fields during movement on a 30 degree conic space are illustrated in the right column of Fig. 3. It can be noted that in case of all four of these solids, the resultant grid fields are not hexagonal, unlike the case of hexagonal column. However, for all of these solids, grid fields remain very much hexagonal when the amount of tilt is only a few degrees (details not shown here). This corresponds well to the usual experimental conditions where the horizontal plane may include some small amount of tilt by chance.

Table 1. Property of several polyhedrons that we consider possible shapes of 3D grid fields

Solid name	Direction vectors	Number of faces	Properties
Hexagonal prism	$(0,90),(60,90),(120,90)$ $(0,0)$	8 	Space filling, every layer is same along Z-axis
Hexagonal bipyramid	$(0,60),(60,60),(120,60),$ $(180,60),(240,60),(300,60)$	12 	Every layer along the Z-axis is same
Combination of Tetrahedron and Octahedron	$(330.05,116.52),(90,116.525),$ $(29.95,63.48),(0,180)$	4 8 	Combination is space filling, and every alternate layers are same
Rhombic hexahedron	$(0,60),(60,120),(120,60),$ $(180,120),(240,60),(300,120)$	6 	Space filling, every third layer is same

The simulation results showed in Fig. 3 suggest that though the grid fields we observe during horizontal movement are hexagonally formed, they may not be so when the plane of movement is tilted. In earlier experiments [8][9], it was found that remapping of place fields occur when rats move on either tilted space or a three-dimensional environment like the space shuttle. As grid cells provide the major input to place cells, we can predict that in considerable amount of tilted space or in 3D space, the resultant grid fields may be of different shapes compared to 2D horizontal, causing remapping of place cells. However, due to the lack of experimental proof, we cannot single out any one of the above mentioned solids as the most probable shape of 3D grid field. We believe that experiments on grid cell recording during rats movements on tilted or curved space may reveal the true shape of the three dimensional grid fields, if they exist at all.

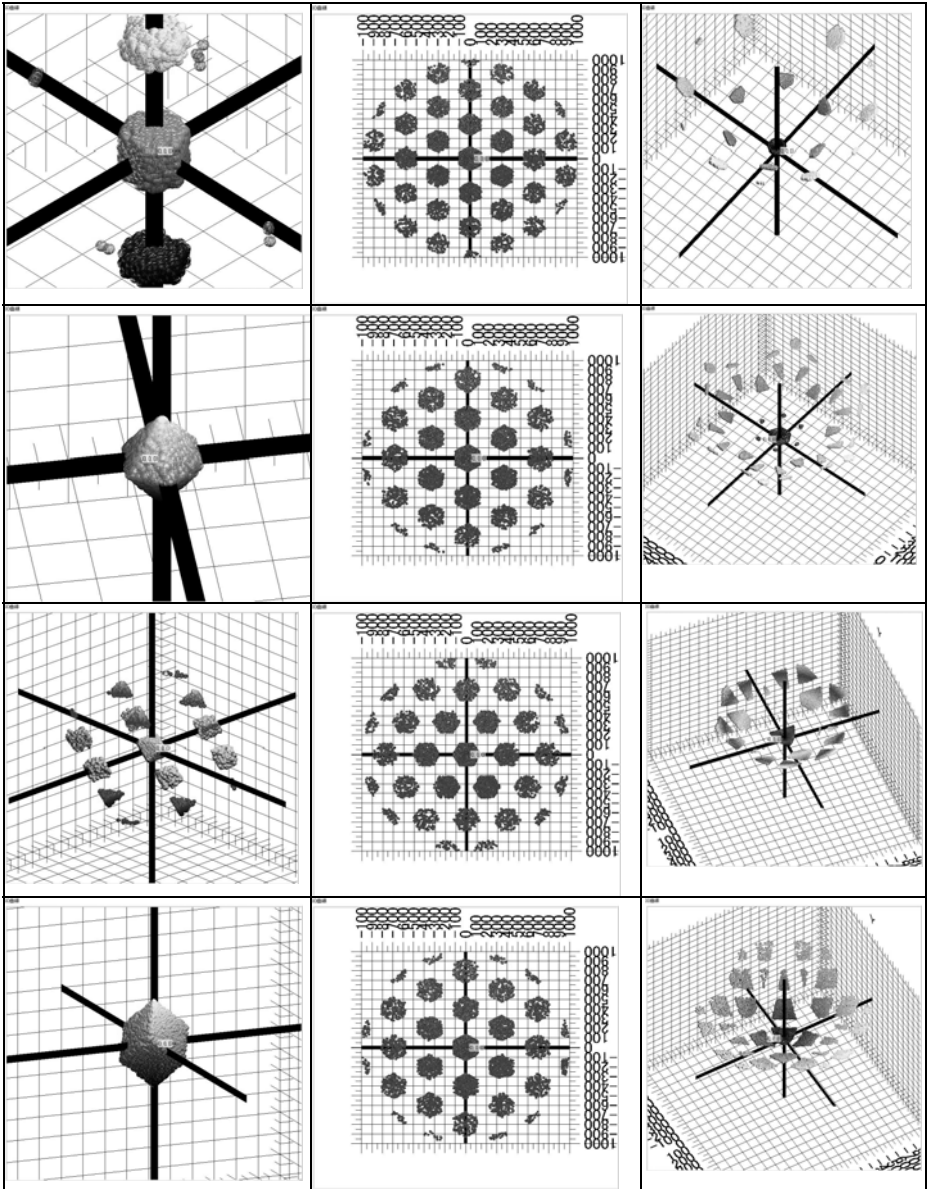


Fig. 3. Simulation results of our 3D grid model for hexagonal prism, hexagonal bipyramid, tetrahedron-octahedron combo, and rhombic hexahedron. Left column: 3D grid fields in the shapes of these solids generated by our 3D model, with appropriately chosen direction vectors. Note that only one grid field in each case is shown for clarity. Middle column: All these polyhedrons generate hexagonal grid fields when tilt angle is 90 degrees. These fields correspond well with the experimentally found grid fields. Right column: Grid fields for the same polyhedrons during simulated movement on a 30 degree conic space.

4 Discussion and Conclusion

In this paper, we made some predictions about the possibility of three-dimensional properties of grid cells. It should be noted that by “three-dimension” we mean space that is not a horizontal plane. Therefore, a tilted conic space (used in the related simulations) or a slope is three-dimensional in space in our definition because they are not horizontal planes. We assumed that the animal’s sense of a ‘tilt-angle’, which can be relayed from sensory system to EC, is additionally required in our extended model of three-dimensional grid fields. With the assumption of the existence of three-dimensional grid fields, we have showed some simulation results of possible 3D grid fields of various shapes, considering various possibilities of their layered formation along the 3D space. It is also shown that in case of movement in horizontal plane, our three-dimensional model is reduced to the original two-dimensional model to generate hexagonal grid fields. Without much experimental data to support our hypothesis, and given that the existence of 3D grid fields are yet to be found, we cannot conclude any one of these polyhedral shapes as the most probable one, but we believe our model can be helpful to make some predictions on the characteristics of EC grid cells in future.

Acknowledgements. This work was supported by Japanese Government KIBAN-S KAKENHI (20220003).

References

1. O’Keefe, J., Dostrovsky, J.: The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Res.* 34, 171–175 (1971)
2. O’Keefe, J.: Place units in the hippocampus of the freely moving rat. *Exp. Neurol.* 51, 78–109 (1976)
3. O’Keefe, J., Nadel, L.: *The Hippocampus as a Cognitive Map*. Clarendon, Oxford (1978)
4. Muller, R.U., Kubie, J.L., Ranck Jr., J.B.: Spatial firing patterns of hippocampal complex-spike cells in a fixed environment. *Journal of Neuroscience* 7, 1935–1950 (1987)
5. Jung, M.W., McNaughton, B.L.: Spatial selectivity of unit activity in the hippocampal granular layer. *Hippocampus* 3, 165–182 (1993)
6. Hafting, T., Fyhn, M., Molden, S., Moser, M.B., Moser, E.I.: Microstructure of a spatial map in the entorhinal cortex. *Nature* 436, 801–806 (2005)
7. McNaughton, B.L., Battaglia, F.P., Jensen, O., Moser, E.I., Moser, M.B.: Path integration and the neural basis of the “cognitive map”. *Nature Rev. Neuroscience* 7, 663–678 (2006)
8. Knierim, J., McNaughton, B.L., Poe, R.: Three-dimensional spatial property of hippocampal neurons during space flight. *Nature Neuroscience* 3, 211–212 (2000)
9. Knierim, J., McNaughton, B.L.: Hippocampal place-cell firing during movement in three-dimensional space. *Journal of Neurophysiology* 85, 105–116 (2001)
10. Burgess, N.: Grid cells and theta as oscillatory interference: Theory and predictions. *Hippocampus* 18, 1157–1174 (2008)
11. Islam, T., Yamaguchi, Y.: Representation of an Environmental Space by Grid Fields: A Study with a Computational Model of the Grid Cell Based on a Column Structure. In: *Proceedings of IJCNN 2009* (2009)

Data Modelling for Analysis of Adaptive Changes in Fly Photoreceptors

Uwe Friederich^{1,2}, Daniel Coca¹, Stephen Billings¹, and Mikko Juusola²

¹ University of Sheffield, Department of Automatic Control and Systems Engineering,
Mappin Street, Sheffield S1 3JD

² Department of Biomedical Science, Western Bank, Sheffield S10 2TN, UK
{u.friederich,d.coca,s.billings,m.juusola}@shef.ac.uk
<http://www.shef.ac.uk/acse>

Abstract. Adaptation is a hallmark of sensory processing. We studied neural adaptation in intracellular voltage responses of the R1-R6 photoreceptors, of the fruit fly *Drosophila*, subjected to light patterns of naturalistic distribution at varying intensity levels. We use experimental data in a stepwise empirical modelling procedure to estimate a non-linear dynamical model (NARMAX) with variable gain. This model can describe accurately the observed adaptation process at each new level of changing light inputs. Generalized frequency response functions were used to visualize and quantify adaptation in the frequency domain.

Keywords: Non-linear system identification; NARMAX; Generalized frequency response functions; Neural adaptation; Gain adaptation; *Drosophila*; Naturalistic stimulation.

1 Introduction

Adaptation enables efficient encoding of sensory information in single neurons or neural chains. This it does by tuning the system's input-output relationship so that the neural output can best represent sensory information [1,2,3,4]. For example, although light intensity in a natural scene can vary thousand-fold [5], photoreceptors have no difficulties in encoding these patterns. Despite their limited dynamic range¹, photoreceptors can discriminate contrast over the full extend of light levels [5]. Because our understanding of the underlying physiological processes of phototransduction is limited, so are our biophysical models. Therefore, empirical modelling methods have a great value in comprehending the system's overall neural functions and in producing hypothetical models that can be tested experimentally.

Starting with the pioneering work of Marmarelis and McCann in the early 1970s [6,7], various authors have applied non-linear system analysis to study dynamics in early visual neurons. The most common approach has been the identification of Volterra kernels based on the Cross-Correlation method [8,9] or the sum-of-sinoids method [10,11]. Both methods have been strongly restricted in the selection of the stimuli, to be either a mixture of sinoids or Gaussian White Noise (GWN). The latter has been

¹ Dynamic range: Here defined as the ratio of the maximum response and the noise level.

shown to linearise fly photoreceptor outputs and does not excite its nonlinear dynamics as natural scenes do [12,15]. Motivated by this observation, van Hateren developed a model that is able to simulate fly photoreceptor responses to natural light statistics [13]. However, in this study, the light stimuli was limited to a 2-3 log unit range and the model itself had a fixed structure. To study adaptive mechanisms, a more flexible model structure is desirable. Whilst new kernel based methods [14,15] are not restricted in the input distribution anymore, large training data sets are still required for the estimation of higher order kernels.

To overcome difficulties encountered in previous studies, we employ a well established nonlinear system identification methodology developed for NARMAX (Nonlinear Auto Regressive Moving Average with eXogenous inputs) models. The estimation of a parametric NARMAX model requires a minimum on theoretical assumptions and is therefore as flexible as kernel or neural network based methods. Apart from that, the clear and concise parametric model structure allows systematic analysis of the underlying system dynamics. The modest number of NARMAX model parameters can be reliably estimated from small data sets, independent of input data statistics. As such, the NARMAX model allows to track and analyze neural adaptation to give new insight into coding strategies of sensory neurons. Once a model has been identified, it can be analytically transformed into generalized frequency response functions (GFRF). The combined approach allows the study of the system dynamics in both, the time and the frequency domain. Analysis on GFRF provide a tool for studying how adaptation changes the frequency dependent interactions between the input and output.

Based on the NARMAX methodology, we estimated models that can accurately predict photoreceptors' voltage responses to temporal light patterns of naturalistic distribution [5]. Individual models were estimated for light levels ranging in logarithmic steps 10,000 fold. Analysis on GFRF allowed us to find a combined model structure and a single parameter set to approximate adaptive changes by a pure change in the input gain.

The data for this study has been acquired from the "small" fly, *Drosophila*, rather than from previously used "big" flies to make use of its extensive genetic and molecular toolbox [16]. Targeted manipulation at each neural layer of the flies visual system will allow us in a later stage of this study to obtain more insight which neural interactions (*e.g.* lateral synaptic connections, feedback from higher order cells, etc.) influence adaptation and how.

2 Methodology

2.1 Measurements and Stimuli

Wild-type Canton-S strains of *Drosophila* were used in the experiments. The flies were prepared *in vivo* as in [17]. Intracellular voltage responses of blue-green-sensitive R1-R6 photoreceptor cells were recorded using sharp quartz microelectrodes. Photoreceptors were excited by a point of light at the center of their receptive field, as delivered through liquid light guides, connected to high performance LEDs (Fig 3(a)). The measured linear light output of the LEDs was taken as the input to the photoreceptors. Light

patterns were selected from the van Hateren’s natural-stimulus-collection [5]. The stimuli was played back at 2 kHz and measured by a photo diode circuit. Voltage responses (output) and light stimuli (input) were low-pass filtered with a cutoff at 1 kHz before sampling with 2 kHz, and stored for off-line analysis. Light input was attenuated by neutral density filters. This attenuation was performed very rapidly (< 0.1 ms) during the experiments (Fig. 1). To test the range of adapting inputs, the same temporal light pattern was shown to the fly with 0, 1, 2, 3 and 4 log intensity units attenuation, allowing 5 different adaptive levels, named as BG0-BG4; BG0 = very bright, BG4 = very dim. Stimulation at each light level lasted for 20s (Fig. 1). Within this time, a 2s pattern was repeated for us to quantify data variation.

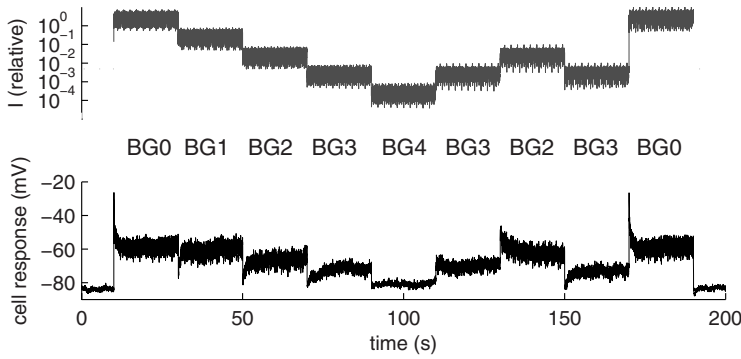


Fig. 1. (top) Relative light intensity values (top) and corresponding photoreceptor responses (bottom). Sections of individual light levels are marked as BG0 (very bright) to BG4 (very dim).

2.2 Data Pre-processing

For system identification, it is necessary that the bandwidth of the excitatory signal does not extensively exceed that of the system under study [18]. It has been shown before that for white noise stimulation, *Drosophila* photoreceptors can follow inputs with $\lesssim 100$ Hz [17]. However, naturalistic stimuli evokes larger responses and might extend the bandwidth [19]. For this reason the input and output data sequences were pre-filtered by a Butterworth low-pass filter with a 200 Hz cut-off and resampled to 400 Hz [18].

2.3 Signal to Noise Ratio

Neural recordings are generally noisy because biochemical reactions are probabilistic rather than deterministic processes. Additionally, recordings from *Drosophila* photoreceptors are particularly sensitive to measurement noise, because the tiny cell dimensions make stable recordings difficult. Moreover, at low light intensity stimulation, photon shot noise has a significant influence on the stimuli and therefore indirectly effects the photoreceptor outputs [3].

The quality of modelling directly depends on the noise level in the data. Therefore, we quantify the Signal-to-Noise Ratio (SNR) of the output for each input light level.

Because the number of repetitions is limited, we apply an bias corrected SNR estimation procedure [20][13]. We estimate the signal $y_{raw} = \bar{y}$ by the ensemble average as $\bar{y} = \frac{1}{J} \sum_{i=1}^J y_i(t)$ from measured responses y_i , $i = 1 \dots J$ to J repeated stimuli. Adopting the notation in [13], we obtain the raw signal and noise power in the voltage output by

$$P_{S_{raw}} = \frac{1}{T} \int_0^T \bar{y}^2(t) dt \quad \text{and} \quad P_{N_{raw}} = \frac{1}{J} \sum_{i=1}^J \frac{1}{T} \int_0^T (y_i(t) - \bar{y})^2(t) dt \quad , \quad (1)$$

and the bias corrected signal and noise power estimate by

$$\hat{P}_S = P_{S_{raw}} - \frac{1}{J} \hat{P}_N \quad \text{and} \quad \hat{P}_N = \frac{N}{N-1} P_{N_{raw}} \quad . \quad (2)$$

Thus, the here applied SNR estimate is given by the ratio of the bias corrected signal power over the power of noise

$$SNR = \frac{\hat{P}_S}{\hat{P}_N} \quad . \quad (3)$$

2.4 NARMAX Modelling Methodology

NARMAX is a methodology to estimate and validate nonlinear difference equation models purely from observations of a system's response to its environmental stimuli. Since the introduction of the NARMAX model by Billings and Leontaritis [21][22], it has been successfully applied in the identification and analysis of a wide range of engineering, biomedical and financial systems. The NARMAX model, is given as

$$y(t) = f(y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u), e(t-1), \dots, e(t-n_e)) + e(t) \quad , \quad (4)$$

where $y(t)$, $u(t)$ and $e(t)$ are the sampled system output, input and error sequences, respectively. $f(\cdot)$ is the nonlinear mapping vector; n_y , n_u and n_e are the maximum lags in the output, input and noise. The noise variable $e(t)$ is a zero mean independent sequence, which accommodates the effects of measurement noise, modelling errors and unmeasured disturbances. $e(t)$ is often referred to as the prediction error, which is defined as $e(t) = y(t) - \hat{y}(t)$, where $\hat{y}(t)$ is the one step ahead prediction of $f(\cdot)$. In this study, $f(\cdot)$ has a l -order polynomial structure with a single input and output (SISO), such that equation (4) becomes

$$y(t) = \theta_0 + \sum_{i_1=1}^n \theta_{i_1} x_{i_1}(t) + \sum_{i_1=1}^n \sum_{i_2=i_1}^n \theta_{i_1, i_2} x_{i_1}(t) x_{i_2}(t) + \dots \quad (5)$$

$$\dots + \sum_{i_1=1}^n \dots \sum_{i_l=i_{l-1}}^n \theta_{i_1, \dots, i_l} x_{i_1}(t) \dots x_{i_l}(t) + e(t) \quad ,$$

where $x(t)$ denotes the lagged variables in y , u and e . n is the sum of variables $n = n_y + n_u + n_e$, and θ_i are scalar parameters to be estimated. Hammerstein, Wiener, Bilinear and Volterra models that have been previously applied to model neural systems are all subclasses of the polynomial NARMAX model and can be derived from (5) [23]. The prediction error terms $e(\cdot)$ are included in the NARMAX model to accommodate

noise. Although in this paper we apply no further analysis on the noise model, it is estimated to ensure the process model is unbiased.

In the here applied method, we employ parameter estimation, structure detection and model validation in an interlinked procedure [24].

- **Term Selection and Parameter Estimation by the OLS Algorithm.** The model structure (5) is linear in its parameters θ_i , this allows the construction of a linear regression model in matrix form,

$$y(t) = \sum_{i=1}^M p_i(t)\theta_i + \epsilon(t), \quad t = 1 \dots N \quad \text{or} \quad \mathbf{y} = \mathbf{P}\Theta + \epsilon, \quad (6)$$

where N denotes the length of the training data set, the $p_i(t)$ are monomials (terms) of $x_1(t)$ to $x_n(t)$ up to degree l . The modelling error sequence ϵ is iteratively obtained. $\Theta = [\theta_0, \dots, \theta_{i_1, \dots, i_l}]^T$ is the M -dimensional parameter vector to be estimated.

Under the condition that \mathbf{P} has full rank, the Orthogonal Least Squares (OLS) algorithm [25] applies an orthogonal decomposition of the regression matrix, such that $\mathbf{P} = \mathbf{W}\mathbf{A}$, with \mathbf{W} being an orthogonal matrix, satisfying $\mathbf{D} = \mathbf{W}^T\mathbf{W}$, where \mathbf{D} is $\text{diag}(d_1, \dots, d_M)$. Equation (6) is therefore equivalent to $\mathbf{y} = \mathbf{W}\mathbf{g} + \epsilon$, with $\mathbf{g} = \mathbf{A}\Theta$. Instead of estimating Θ directly, $\hat{\mathbf{g}}$ is estimated as the linear least squares solution that minimizes $\|\mathbf{y} - \mathbf{W}\mathbf{g}\|$, where $\|\cdot\|$ is the euclidean norm. \mathbf{W} being orthogonal allows to calculate each element g_i (for the i^{th} term) in $\hat{\mathbf{g}}$ independently. The error reduction ratio $ERR_i = \frac{g_i^2 d_i}{\mathbf{y}^T \mathbf{y}}$ is used to evaluate for each g_i , how much the corresponding term contributes to the output. In a forward regression manner, terms are chosen first that contribute more to the output until the selection is stopped, when all significant terms are selected. In general only a small number of terms $m \ll M$ is enough for approximating the systems dynamics [25]. Eventually, the parameter estimates are calculated from $\hat{\Theta} = \mathbf{A}^{-1}\hat{\mathbf{g}}$.

- **Model Validation.** Cross validation was used to evaluate the model performance on unseen data. This ensures that the model describes the underlying dynamical process and not just the training data. Models are selected and validated, based on the performance in the following complementary tests.

- *Normalized Mean Square Error (NMSE)*

$$NMSE = \frac{\sum_{t=1}^{N_d} (\hat{y}(t) - y(t))^2}{\sum_{t=1}^{N_d} (y(t) - E[y])^2}, \quad (7)$$

where $E[y] = \frac{1}{N_d} \sum_{t=1}^{N_d} y(t)$ is the mean of the measured output, N_d is the validation data length and $\hat{y}(t)$ are the model predictions. Depending, if the predictions are purely model predicted outputs, $\hat{y}(t) = \hat{y}(t)_{MPO}$ or one step ahead predictions $\hat{y}(t) = \hat{y}(t)_{OSA}$, the $NMSE_{MPO}$ or the $NMSE_{OSA}$ are evaluated.

- *Final Prediction Error (FPE)* [26]

$$FPE = \frac{N_d + m\gamma}{N_d - m\gamma} \sigma_e^2, \quad (8)$$

where σ_e^2 is the variance of the error sequence (residuals) $e(t) = y(t) - \hat{y}(t)$, $t = 1 \dots N_d$ and m is the number of selected terms. The measure is used to reduce the spread of the error penalized by the model size.

- *Higher order correlation tests*

Residuals of adequately estimated nonlinear models should be uncorrelated with all linear and non-linear combinations of past inputs and outputs, they should be unpredictable and thus contain no information about the dynamics of the system. This is tested by the following two higher order correlation tests, [27],

$$\begin{aligned} \Phi_{(\epsilon^2)'(\mathbf{y}\epsilon)'}(\tau) &= \kappa \delta(\tau) \quad \forall \tau \\ \Phi_{(\mathbf{u}^2)'(\mathbf{y}\epsilon)'}(\tau) &= 0 \quad \forall \tau, \end{aligned} \quad (9)$$

where κ is a constant $0 < \kappa < 1$ and $\delta(\cdot)$ is the delta function. Φ_{vw} is the normalized correlation function of signal v and w . The dash, $(\cdot)'$, denotes that the mean level of the signal in brackets is removed. Shifts between v and w are selected as $\tau = -100 \dots 100$. The correlations are never exactly zero for all lags and the 95% confidence bands, defined as $\pm \frac{1.96}{\sqrt{N_d}}$, are used to indicate, if the estimated correlations are significant or not [27].

Estimation and validation algorithms have been implemented in MATLAB[®], allowing a consistent modelling approach for all models, part of this study. Although, all models are validated by each of the described measures, for clarity, in the results section only the NMSE is shown.

2.5 Generalized Frequency Response Functions

The NARX model, a subset of the NARMAX, containing functionals of lagged inputs and outputs alone, can (under some assumptions) be expanded into a Volterra functional polynomial of the input $u(t)$ only [28]. The discrete Volterra Series is defined as, [29],

$$y(t) = \sum_{n=1}^{\infty} y_n(t), \quad (10)$$

where $y_n(t)$ denotes the n -th order output of the system and is given by,

$$y_n(t) = \sum_0^t \dots \sum_0^t h_n(k_1, \dots, k_n) \prod_{i=1}^n u(t - k_i), \quad (11)$$

where $h_n(k_1, \dots, k_n)$ is called the 'n-th order kernel' or the 'n-th order impulse response' of the system. The multidimensional Fourier transform of the n -th order impulse response $h_n(k_1, \dots, k_n)$ yields the n -th order transfer function or the 'n-th order Generalized Frequency Response Function (GFRF)

$$H_n(j\omega_1, j\omega_2, \dots, j\omega_n) = \sum_{\omega_1=-\infty}^{\infty} \dots \sum_{\omega_n=-\infty}^{\infty} h_n(k_1, \dots, k_n) e^{-j(\omega_1 k_1 + \dots + \omega_n k_n)}. \quad (12)$$

The first order GFRF, $H_1(j\omega)$ explains linear effects, while the nonlinear GFRF's, $H_n(j\omega_1, \dots, j\omega_n)$ $n > 1$, give a measure of the nonlinear coupling of the input spectral components and reveal energy transfer mechanisms to new spectral components in the output [30].

For this study, GFRF have been analytically computed directly from identified discrete time NARX models, applying the recursive algorithm developed by Peyton Jones and Billings [31]. In contrast to the direct estimation of GFRF from input output data, this method requires significantly less data samples.

3 Results

We measure voltage responses in *Drosophila* photoreceptors to naturalistically distributed light contrast time series (Fig. 3(a)). The same light pattern was repeated at different light levels (BG0-BG4) to obtain a running account, how adaptation dynamics change with illumination. BG0 is the brightest level; BG1 gives the same pattern but 10-times less intense; BG2 is 100-times weaker than BG0, etc (Fig. 1). From corresponding light input and photoreceptor voltage output, NARMAX models were estimated and mapped into the frequency domain as GFRFs. Data analysis and model identification are implemented as a four step procedure.

Step 1: SNR estimation of voltage outputs at each BG level.

Step 2: Identification of local NARMAX models, at each BG level separately (M_{BG0} to M_{BG3}).

Step 3: Identification of a global model structure that can explain the complete data set by adjusting its parameters ($M_G(\Theta_{BG1})$ to $M_G(\Theta_{BG3})$).

Step 4: Identification of a global model that can explain the complete data set by adjusting only its input gain, α ($M_G(\Theta_G, \alpha_1)$ to $M_G(\Theta_G, \alpha_3)$).

Models are estimated from training data sets, containing 800 input/output samples. All shown NMSE values are based on 6400 output predictions, simulated by models that performed best in all validation tests. These values are used as a performance index to compare models.

3.1 Data Variability Analysis

For each BG level, the SNR (3) has been calculated from voltage responses to $J = 8$ input repetitions. The results, summarized in Table 1 show a significant decay of signal Power \hat{P}_S in comparison to noise Power \hat{P}_N , as the light intensity decreases. Consequently, the SNR values drop in the same manner. At lower light intensities, less

Table 1. Signal power, noise power Signal to Noise Ratio (3) at distinct BG levels

	BG0	BG1	BG2	BG3	BG4
\hat{P}_S	9.13	10.02	7.66	2.71	0.247
\hat{P}_N	0.37	0.53	0.69	0.80	0.68
SNR	24.91	18.87	11.03	3.40	0.36

photons are available to activate the phototransduction cascade, which leads to smaller voltage responses [32]. At the same time stochastic photon capture in the photoreceptors induces additional randomness and decreases the SNR at lower light levels. This trend resembles the results shown previously in [33]. Because noise in the input or in the output cannot be simulated, model predictions deviate from output measurements, even if a model captures perfectly the underlying system dynamics. This has a direct implication on any prediction error based validation test. Therefore, data with low SNR values inevitably lead to higher NMSE values. For this reason models estimated from input/output data at dimmer light levels inevitably show poorer performance than models at bright light intensities.

3.2 Individual Model Estimation at Each BG Level

In this part of the study, the structure and parameters of NARMAX models were individually estimated from stimuli-response data BG0 to BG3 (cf. Fig. 1). For BG4, no reliable model could be found. The low $SNR = 0.558$ suggests that for this dim inputs individual photoreceptors cannot discriminate light patterns anymore from noise and produce mostly random quantum fluctuations [17]. Table 2 contains models M_{BG0} (bright input) to M_{BG3} (dim input) and their performance index ($NMSE$). NMSE values show that models estimated from responses to brighter inputs (M_{BG0} to M_{BG2}) predict remarkably well throughout the same light level, even for data sets that were not used for estimation. The significant poorer performance of the M_{BG3} model is a consequence of the low SNR at the dim light level.

Throughout all the tested light levels, second order polynomial models are sufficient to model the observed dynamics of the underlying nonlinear system. Higher order polynomial models were also investigated, but these did not improve the model performance. Despite the input changes by 3 log units, the structures of models M_{BG0} to M_{BG3} (Table 2) are very similar. Terms in models of different BG levels vary mostly in $+/-$ one lag. The strong similarity in the set of detected terms for models of different light levels suggests that a global model structure can explain the data for all tested input levels.

3.3 Parameter Estimated Models with Constant Structure

Various combinations of terms in Table 2 were tested to construct a global model structure that performs well at all BG levels. The best structure was found to be the previously detected M_{BG0} term set. Adopting the structure from M_{BG0} , the global structure is called $M_G(\Theta_{BGi})$ with $\Theta_{BGi} = [\hat{\theta}_0^{BGi}, \dots, \hat{\theta}_{14}^{BGi}]$, $i = 0, \dots, 3$, being the model parameters estimated individually at light levels BG0-3. Table 3 summarizes the results for individually estimated parameter sets, from input/output data at light levels BG0-3.

At all tested light levels, the model performance does not decrease for keeping a global model structure, compared to values of models with individually estimated structures. For dimmed inputs, parameter estimated models with a constant structure even perform slightly better. This suggest that a single nonlinear model with varying parameters indeed can be used to describe the input-output data set.

Table 2. Independently estimated NARMAX models for different light levels. Model parameters are presented in columns. The first column contains corresponding terms for each parameter. “-” denotes that a term is not part of the model. The last column contains the global model structure $M_G(\Theta)$.

<i>terms</i> \ <i>models</i>	M_{BG0}	M_{BG1}	M_{BG2}	M_{BG3}	$M_G(\Theta)$
offset	-58.42	-59.75	-66.39	-72.2423	
c	-2.026	-2.309	-1.792	-0.616	$\hat{\theta}_0$
y(t-1)	0.964	1.009	1.089	0.949	$\hat{\theta}_1$
y(t-2)	-	-0.154	-0.209	-	
y(t-3)	0.173	-	-	-	$\hat{\theta}_2$
y(t-4)	-0.348	-	-	-	$\hat{\theta}_3$
y(t-5)	0.093	-	-	0.135	$\hat{\theta}_4$
u(t-4)	0.165	1.551	17.15	-	$\hat{\theta}_5$
u(t-5)	0.279	3.857	27.06	-	$\hat{\theta}_6$
u(t-6)	0.257	3.523	21.90	132.83	$\hat{\theta}_7$
u(t-7)	0.126	-	-	-	$\hat{\theta}_8$
y(t-1)u(t-5)	-	-	-1.319	37.58	
y(t-2)u(t-4)	-0.030	0.311	-	-12.778	$\hat{\theta}_9$
y(t-2)u(t-5)	-	-	-	-38.5616	
y(t-5)u(t-4)	0.051	0.671	-	-	$\hat{\theta}_{10}$
y(t-5)u(t-5)	-	-	0.487	-	
y(t-6)u(t-4)	-0.039	-0.635	-0.881	-	$\hat{\theta}_{11}$
u(t-3)u(t-7)	0.012	-	-	-	$\hat{\theta}_{12}$
u(t-4)u(t-5)	-0.015	-0.399	-87.15	-	$\hat{\theta}_{13}$
u(t-5)u(t-6)	-	-4.420	-89.13	-8602	
u(t-6)u(t-7)	-0.028	-	-	2466	$\hat{\theta}_{14}$
$NMSE_{MPO}$	0.094	0.083	0.096	0.254	
$NMSE_{OSA}$	0.016	0.018	0.027	0.067	

Table 3. Performance of models with global structure and BG-dependent parameter estimates

Uni Model	$M_G(\hat{\Theta}_{BG0})$	$M_G(\hat{\Theta}_{BG1})$	$M_G(\hat{\Theta}_{BG2})$	$M_G(\hat{\Theta}_{BG3})$
$NMSE_{MPO}$	0.094	0.067	0.096	0.23
$NMSE_{OSA}$	0.016	0.017	0.025	0.064

To investigate adaptative changes in the frequency domain, the first and second order GFRF's $H_{1,BG_i}(j\omega)$ and $H_{2,BG_i}(j\omega_1, j\omega_2)$, $i = 0..3$ were computed for the identified NARMAX models $M_G(\Theta_{BG0})$ to $M_G(\Theta_{BG3})$, respectively. Fig. 2 summarizes the results in plots of the first-order functions (Fig. 2(a)) and selected slices through second-order functions (Fig. 2(c), 2(d)). The location of slices is shown in Fig. 2(b). Analysis on the first and second order GFRF magnitude plots in Fig. 2 reveal that the 3dB bandwidth of the system remains constant, at about 20 Hz, regardless of the light level. An energy transference phenomenon like described in [30] is not eminent. For decreasing intensity levels, the magnitude curves are shifted upwards whilst their

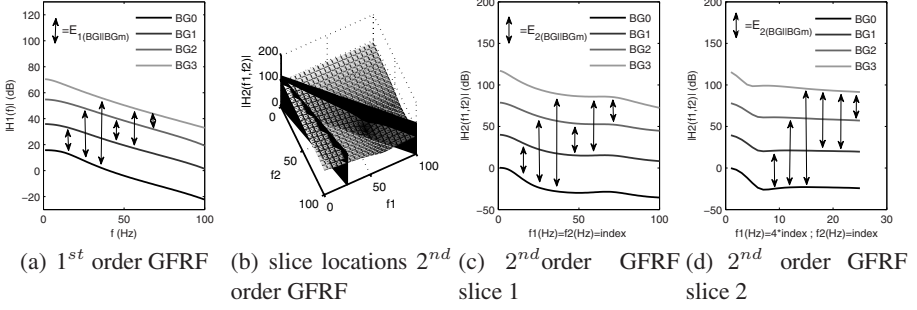


Fig. 2. 1st order GFRF plots of and slices through 2nd order GFRF for models $M_G(\Theta_{BG1})$ to $M_G(\Theta_{BG3})$. Changes between frequency responses of different light levels are indicated by arrows.

shape remains almost the same. This suggests that the photoreceptor adaptation to lower light intensities is manifested just through an increase of the input gain. If this hypothesis is correct then we would expect that the second order magnitude plots will be shifted upwards by an amount equal to the squared linear shift.

Indeed, assuming the Volterra Series in (10) is expanded up to the second order kernel, then its Fourier transform yields, (34),

$$Y_1(j\omega_1, j\omega_2) = H_1(j\omega_1)U(j\omega_1) + H_2(j\omega_1, j\omega_2)U(j\omega_1)U(j\omega_2) , \quad (13)$$

where $H_1(\cdot)$ and $H_2(\cdot)$ are the first and second order GFRF in (12) and $Y(\cdot)$ and $U(\cdot)$ are the Fourier Transforms of the output and input, respectively. Assuming, the input signal is modified by a constant gain α , then (13) yields,

$$Y_2(j\omega_1, j\omega_2) = H_1(j\omega_1)\underline{\alpha U(j\omega_1)} + H_2(j\omega_1, j\omega_2)\underline{\alpha U(j\omega_1)\alpha U(j\omega_2)} \quad (14)$$

$$\Leftrightarrow Y_2(j\omega_1, j\omega_2) = \underline{\alpha H_1(j\omega_1)U(j\omega_1)} + \underline{\alpha^2 H_2(j\omega_1, j\omega_2)U(j\omega_1)U(j\omega_2)} , \quad (15)$$

where $Y_1(\cdot) \neq Y_2(\cdot)$. The underlines in equations (14) and (15) highlight that in a second order Volterra Model, a change in the input signal by a constant gain α is equivalent to a constant gain α in $H_1(\cdot)$ and a quadratic gain α^2 in $H_2(\cdot)$. To test, if this is the case for changes in GFRF's of models $M_G(\Theta_{BG0})$ to $M_G(\Theta_{BG3})$, we calculated α as the arithmetic mean $E_{BGl|BGm}$ of shifts between first order GFRF curves $H_{1,BGl}(\cdot)$ and $H_{1,BGm}(\cdot)$, $l, m = 0..3$ as

$$E_{1(BGl|BGm)} = \frac{1}{\omega_{max}} \sum_{\omega=1}^{\omega_{max}} H_{1,BGl}(j\omega) - H_{1,BGm}(j\omega) . \quad (16)$$

These, we compared to corresponding mean shifts between second order GFRF surfaces calculated as

$$E_{2(BGl|BGm)} = \frac{1}{\omega_{max}^2} \sum_{\omega_1=1}^{\omega_{max}} \sum_{\omega_2=1}^{\omega_{max}} (H_{1,BGl}(j\omega_1, j\omega_2) - H_{2,BGm}(j\omega_1, j\omega_2)) , \quad (17)$$

where $H_{2,BGi}(\cdot)$ denotes the second order GFRF of model $M_G(\Theta_{BGi})$, $i = 0 \dots 3$. According to previous argumentation, for a pure change in input gain, $E_{2(BGl|BGm)} \approx \alpha^2$ needs to be satisfied for all combinations of $l, m = 0 \dots 3$. To test this, we transformed the second order shift into α^p with $p = \frac{\log(E_{1(BGl|BGm)})}{\log(E_{2(BGl|BGm)})}$. As a measure, how much curves deviate from being a pure shift, we additionally calculated the variance between differences of GFRFs as

$$\sigma_{1(BGl|BGm)}^2 = \frac{1}{\omega_{max}} \sum_{i=1}^{\omega_{max}} (H_{1,BGl}(j\omega) - H_{1,BGm}(j\omega) - E_{BGl|BGm})^2 \quad \text{and} \quad (18)$$

$$\sigma_{2(BGl|BGm)}^2 = \frac{1}{\omega_{max}^2} \sum_{\omega_1=1}^{\omega_{max}} \sum_{\omega_2=1}^{\omega_{max}} (H_{1,BGl}(j\omega_1, j\omega_2) - H_{2,BGm}(j\omega_1, j\omega_2) - E_{2(BGl|BGm)})^2.$$

Results of the evaluation of shifts between first and second order GFRF functions for $\omega_{max} = \frac{100Hz}{2\pi}$ are summarized in Table 4. The analysis of the 2nd order GFRF magnitudes reveal that indeed the shifts in this case have a quadratic tendency relative to the linear shifts. Only shifts to GFRFs of model $M_G(\Theta_{BG3})$ deviate. The shift between the linear GFRF of $M_G(\Theta_{BG2})$ and its 2nd order one is almost cubic ($\alpha^{2.75}$). There could be two reasons for this deviation. If, the shifts to GFRFs for light level BG3 are accurate then, at very dim light levels the nonlinear contribution the output signal enhances. Alternatively, if the high amount of noise at dim light levels leads to biased parameter estimates, causing corresponding GFRFs to be inaccurate while the system in fact would perform pure gain adaptation. In case of the latter, the system performs a normalization of its frequency response at different input light levels, by holding on to a constant spectral characteristic. By adjusting a gain, it maintains the response amplitude within the limited range of 50 mV, while the same frequencies in the output are kept constant throughout all the tested BG levels.

Table 4. Evaluation of adaptive changes between GFRFs of models $M_G(\Theta_{BGl})$ and $M_G(\Theta_{BGm})$, estimated from data at the l^{th} and m^{th} light level BGl and BGm (arrows Fig 2)

(l, m)	(0, 1)	(0, 2)	(0, 3)	(1, 2)	(1, 3)	, (2, 3)
$\alpha = E_{1(BGl BGm)}$	22.66dB	41.37dB	53.95dB	18.70dB	31.29dB	12.58dB
$\alpha^p = E_{2(BGl BGm)}$	43.81dB	82.31dB	117.0dB	38.50dB	73.17dB	34.67dB
$\sigma_{1(BGl BGm)}^2$	2.05dB ²	1.84dB ²	0.60dB ²	0.10dB ²	0.95dB ²	0.93dB ²
$\sigma_{2(BGl BGm)}^2$	3.91dB ²	4.94dB ²	12.34dB ²	1.38dB ²	10.68dB ²	5.93dB ²
p	1.93	1.99	2.16	2.06	2.33	2.75

3.4 Global Model with Gain Adaptation

Frequency normalization in the form of a pure gain adaptation can be modelled by a global model structure with a constant global parameter set Θ_G and a variable input gain α . Indeed, instead of a shifting the GFRF $H_1(\cdot) \rightarrow \alpha H_1(\cdot)$ and $H_2(\cdot) \rightarrow \alpha^2 H_2(\cdot)$ as in (15), the input can be altered by a constant gain $U(\cdot) \rightarrow \alpha U(\cdot)$ as in (14). The same can be shown in the time domain, considering the inverse Fourier Transform of $\alpha U(j\omega) = \alpha u(t)$. Assuming, there exists a unique transformation between a Volterra-Model and a polynomial NARX-Model [35][28], then it can be shown for a polynomial

NARX model (5) that a change of the input variable by a constant α , such that $u(t) \rightarrow \alpha u(t)$, is equivalent to a shift by α , in its first order GFRF, by α^2 , in its 2nd order GFRF, etc (cf proof in [28]).

Motivated by this finding and results, shown in Table 4, we constructed a global model, to explain the full input-output data set by adapting only one parameter, the input gain α . The global model consists of the global model structure $M_G(\Theta_G) = M_{BG0}$, with the best tested parameter set $\Theta_G = \Theta_{BG0}$ and an adjustable input gain α_i , such that $u(t) \rightarrow \alpha_i u(t)$, where the index “ i ” refers to the i^{th} light level BG_i . $M_G(\Theta_G, \alpha_i)$ denotes the global model. Table 5 summarizes the performance of the global model for predicting the output at each light level BG0-3. The parameters α_i , $i = 0 \dots 3$ have been estimated using the MATLAB[®] implementation of the L-M algorithm. The global Model $M_G(\Theta_G, \alpha_i)$, $i = 0 \dots 3$ performs almost as good at each BG level,

Table 5. Performance of global model with BG-dependent input gain α

Uni Model	$M_G(\Theta_G, \alpha_0)$	$M_G(\Theta_G, \alpha_1)$	$M_G(\Theta_G, \alpha_2)$	$M_G(\Theta_G, \alpha_3)$
α_i	1=0dB	11.80=21.4dB	79.14=38.0dB	290.4=49.3dB
$NMSE_{MPO}$	0.094	0.081	0.106	0.239
$NMSE_{OSA}$	0.016	0.018	0.028	0.07

as if the full set of parameters $\hat{\Theta}$ is estimated independently at each light level (cf Table 3). Although, the 2nd order GFRF calculated from the parameter estimated model $M_{BG3}(\Theta_{BG3})$ was not exactly quadratic, however, forcing a quadratic change by a global model, does not significantly decrease the models performance, measured by the $NMSE$. It is therefore possible that even at very dim light levels, like BG3, the system performs a frequency normalization. This result suggests that the input-output data can, within its limitations at low light levels, be described by the suggested global model with light level dependent adjusted input gain α .

4 Discussion

In this paper, nonlinear system identification and analysis techniques were used to investigate the adaptation of *Drosophila* photoreceptors to different light intensity levels. Instant changes between light levels cause the system to respond in distinct adaptive modes, so as to discriminate light patterns, which can vary 10,000 fold. Such coding occurs reliably within the limited voltage range (50-60 mV) of photoreceptors.

For the first time, a unified nonlinear dynamical model of the photoreceptor was derived that explains adaptation at each level of dynamic light inputs as a simple gain adjustment process. Utilizing nonlinear system identification, the new model is based on experimental measurements of photoreceptor responses to naturalistic stimuli. The use of generalized frequency response functions was instrumental in revealing the underlying mechanism of this type of adaptation. The derived model was validated extensively using data sets recorded for different light levels. The graph shown in Fig. 3(b)

² Note: Adjustments to small variations in the output offset have been applied. Since these small adjustments do not change the results they are not further discussed here.

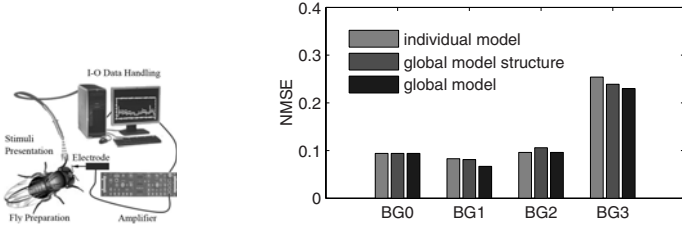


Fig. 3. (a) Experimental Setup; (b) NMSE comparison of applied modelling approaches

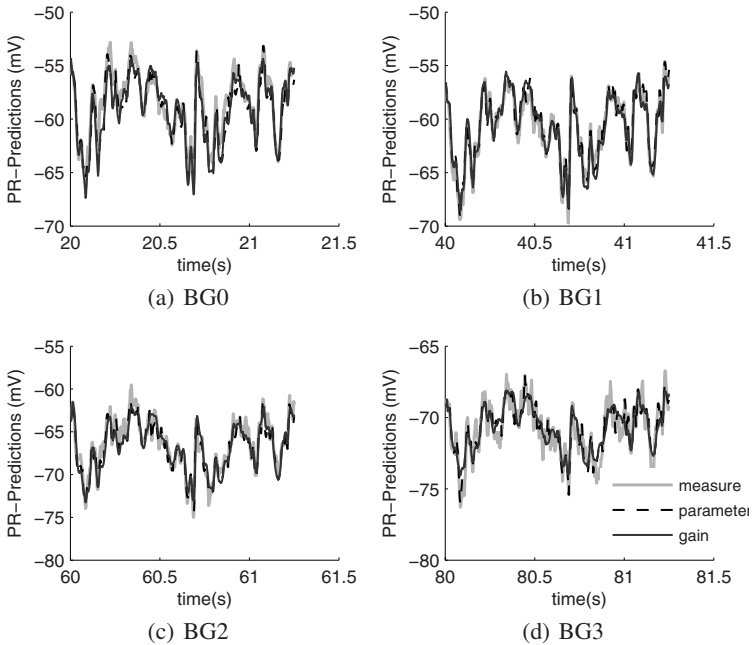


Fig. 4. Evaluation of model predicted outputs \hat{y}_{MPO} at different background light intensities for gain- and parameter adjusted models

summarizes the performance of all the estimated models and highlights that the same model performance could be achieved for individual estimated models, models with fixed structure, and the global model with an adapted gain. The individual model performances are remarkably good when compared to the natural data variation, measured by its *SNR*. Fig. 4 shows the model predictions of parameter and gain adapted models in comparison to the actual recorded voltage responses.

We showed that *Drosophila* photoreceptors adapt to changing light inputs to preserve the spectral structure in its output to higher order neurons. These dynamics are quite different from those shown previously for Gaussian White Noise inputs [17], where the system integrated the dim and differentiated the bright inputs. Instead, when

Drosophila photoreceptors adapt to naturalistic contrasts, it appears that they employ a pure gain control. This was tested by simulating the system with fixed NARMAX model, whilst only optimizing the input gain. These new findings have implications on the understanding how insect eyes code visual information. To learn more about the nature of adaptation, similar experiments, involving visually impaired fly mutants will be carried out. By replacing the constant gain with a variable gain, we will be able to use the derived global model in future studies to investigate the influence of stimulation patterns with different statistics onto adaptation.

Acknowledgement. We thank Hans van Hateren for providing the naturalistic time series of light intensities. This work was supported by the Biological Sciences Research Council (BBF0120711 and BBD0019001 to MJ). DC and SAB gratefully acknowledge that this work was supported by the Engineering and Physical Sciences Research Council and the European Research Council. UF readily acknowledges the support by the University of Sheffield.

References

1. Barlow, H.B.: Possible principles underlying the transformation of sensory messages. MIT Press, Cambridge (1961)
2. Zheng, L., Nikolaev, A., Wardill, T.J., O’Kane, C.J., de Polavieja, G.G., Juusola, M.: Network adaptation improves temporal representation of naturalistic stimuli in drosophila eye: I dynamics. *PLoS ONE* 4, e4307 (2009)
3. van Hateren, J.: A theory of maximizing sensory information. *Biol. Cybern.* 68(1), 23–29 (1992)
4. Wark, B., Lundstrom, B.N., Fairhall, A.: Sensory adaptation. *Sensory systems* 17(4), 423–429 (2007)
5. Van Hateren, J.: Processing of natural time series of intensities by the visual system of the blowfly. *VIS. RES.* 37(23), 3407–3416 (1997)
6. Marmarelis, P.Z., Naka, K.I.: White-noise analysis of a neuron chain: An application of the wiener theory. *Science* 175, 1276–1278 (1972)
7. McCann, G.D.: Nonlinear identification theory models for successive stages of visual nervous systems of flies. *Journal of Neurophysiology* 37, 869–895 (1974)
8. Eckert, H., Bishop, L.: Nonlinear dynamic transfer characteristics of cells in the peripheral visual pathway of flies. part i: The retina cells. *Biological Cybernetics* 17(1), 1–6 (1975)
9. Marmarelis, V., McCann, G.: A family of quasi white random signals and its optimal use in biological system identification. part ii: Application to the photoreceptor of calliphora erythrocephala. *Biological Cybernetics* 27(1), 57–62 (1977)
10. Victor, J., Shapley, R., Knight, B.: Nonlinear analysis of cat retinal ganglion cells in the frequency domain. *Proc. Natl. Acad. Sci. U.S.A.* 74(7), 3068–3072 (1977)
11. Victor, J.: Nonlinear systems analysis: comparison of white noise and sum of sinusoids in a biological system. *Proc. Natl. Acad. Sci. U.S.A.* 76(2), 996–998 (1979)
12. Juusola, M., Kouvalainen, E., Jarvilehto, M., Weckstrom, M.: Contrast gain, signal-to-noise ratio, and linearity in light-adapted blowfly photoreceptors. *Journal of General Physiology* 104(3), 593–621 (1994)
13. Van Hateren, J.H., Snippe, H.P.: Information theoretical evaluation of parametric models of gain control in blowfly photoreceptor cells. *Vision Research* 41(14), 1851–1865 (2001)
14. Marmarelis, V.: *Nonlinear Dynamic Modeling of Physiological Systems*. Wiley Interscience, Hoboken (2004)

15. Korenberg, M., Hunter, I.: The identification of nonlinear biological systems: Volterra kernel approaches. *Ann. Biomed. Eng.* 24(2), 250–268 (1996)
16. Borst, A.: *Drosophila's view on insect vision*. *Current Biology* 19(1) (2009)
17. Juusola, M., Hardie, R.C.: Light adaptation in drosophila photoreceptors: I. response dynamics and signaling efficiency at 25°C. *Journal of General Physiology* 117, 3–25 (2001)
18. Ljung, L.: *System Identification - Theory for the User*, 2nd edn. Prentice Hall, Linköping University, Sweden (1999)
19. Juusola, M., de Polavieja, G.G.: The rate of information transfer of naturalistic stimulation by graded potentials. *J. Gen. Physiol.* 122(2), 191–206 (2003)
20. Mocks, J., Gasser, T., Tuan, P.: Variability of single visual evoked potentials evaluated by two new statistical tests. *Electroencephalogr. Clin. Neurophysiol* 57(6), 571–580 (1984)
21. Billings, S., Leontaritis, I.: Identification of nonlinear systems using parameter estimation techniques, vol. 194, pp. 183–187. IEE Conference Publication, Warwick University (1981)
22. Leontaritis, I.J., Billings, S.A.: Input-output parametric models for non-linear systems. part i: Deterministic non-linear systems; part ii: Stochastic nonlinear systems. *International Journal of Control* 41(2), 303–344 (1985)
23. Pearson, R.K.: *Discrete-Time Dynamic Models*. Oxford University Press, Oxford (1999)
24. Wei, H., Billings, S., Liu, J.: Term and variable selection for non-linear system identification. *International Journal of Control* 77(1), 86–110 (2004)
25. Chen, S., Billings, S.A., Luo, W.: Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control* 50(5), 1873–1896 (1989); Cited By (since 1996): 238
26. Akaike, H.: Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics* 21(1), 243–247 (1969)
27. Billings, S.A., Zhu, Q.M.: Nonlinear model validation using correlation tests. *International Journal of Control* 60(6), 1107–1120 (1994)
28. Diaz, H.: Modeling of nonlinear discrete-time systems from input-output data. *Automatica* 24(5), 629–641 (1988)
29. Volterra, V.: *Theory of functionals and of integral and integro-differential equations*. Blackie, London (1930)
30. Billings, S.A., Tsang, K.M.: Spectral analysis for non-linear systems, part ii: Interpretation of non-linear frequency response functions. *Mechanical Systems and Signal Processing* 3(4), 341–359 (1989)
31. Peyton-Jones, J.C., Billings, S.A.: Recursive algorithm for computing the frequency response of a class of non-linear difference equation models. *International Journal of Control* 50(5), 1925–1940 (1989)
32. Gu, Y., Oberwinkler, J., Postma, M., Hardie, R.C.: Mechanisms of light adaptation in drosophila photoreceptors. *Current Biology* 15(13), 1228–1234 (2005)
33. Zheng, L., De Polavieja, G., Wolfram, V., Asyali, M., Hardie, R., Juusola, M.: Feedback network controls photoreceptor output at the layer of first visual synapses in drosophila. *Journal of General Physiology* 127(5), 495–510 (2006)
34. Chow, T., Hong-Zhou, T., Yong, F.: Nonlinear systems representation. In: *Encyclopedia of Electrical and Electronics Engineering*. Wiley, Chichester (2001)
35. Zhao, X., Marmarelis, V.: Nonlinear parametric models from volterra kernels measurements. *Math. Comput. Model.* 27(5), 37–43 (1998)

A Computational Model of Spatial Imagery Based on Object–Centered Scene Representation

Naoyuki Sato

Department of Complex Systems, School of Systems Information Science, Future University - Hakodate, 116-2 Kamedanakano-cho, Hakodate, Hokkaido 041-8655, Japan

satonao@fun.ac.jp

<http://www.fun.ac.jp/~satonao/>

Abstract. The hippocampus maintains the memory of object–place associations and also produces the ability of a scene expectation at a novel viewpoint. To implement such capabilities, an objects’ distances and directions should be integrated as an allocentric space memory, while its neural dynamics have not been discussed. In this paper, we propose an object–centered scene representation as a component on object–place memory in the hippocampus. By using the representation, an object’s distance and direction at the imagery viewpoint can be calculated as the difference between object–centered and imagery scenes. Moreover, the object–centered scene is applicable for the object–place memory retrieval at the novel location. It is suggested that the object-centered scene representation mediates between egocentric and allocentric space representation and supports the spatial imagery at the voluntary viewpoint.

Keywords: hippocampus, object–place associative memory, frame of reference, spatial cognition.

1 Introduction

The hippocampus is known to maintain the memory of the environment. In the hippocampus, many place cells are selectively activated by a specific portion of the environment, and these are expected to represent a map of the environment. This is called the ‘cognitive map theory’ [1] which is applied to many computational models [2] [3]. The model is further developed [4] with a recent physiological evidence of ‘grid cells’ [5]. In addition to the spatial memory, the hippocampus is further associated with the memory of object–place associations [6] [7] [8] and its theoretical models have also been proposed [9] [10] [11]. Based on a theoretically predicted boundary vector cell (BVC) representation [12], Byrne, Becker and Burgess (2007) [13] proposed a neural network model for memory encoding and storage, retrieval and imagery of the environment. It is unique that this model processes a spatial updating in the mental space. However, the scene computation at novel viewpoints remains unsolved. Some neural representation beyond BVC should be considered.

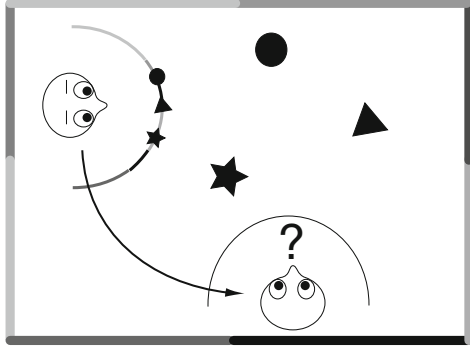


Fig. 1. Object arrangement and scene estimation at an imagery viewpoint

In this paper, we propose an object-centered scene representation as a component of the object-place memory in the hippocampus. Scenes at objects' locations are calculated by a simplified view-based homing algorithm [13], and are applied to the object-scene associative memory encoding, storage, retrieval and imagery. To evaluate the computational ability of the model, the estimated object arrangement at each imagery viewpoints is compared with the object arrangement at the real viewpoint.

2 Model

Figure 2 shows a basic structure of the model consisting of a visual system and a memory system of the hippocampus. This paper focuses on dynamic of object-centered scene transformation and comparison with memorized scenes. The representation of object-centered scene is used to denote object location in the environment and to produce a method for calculating distance and orientation of a voluntary object pair.

2.1 Visual Environment and Visual Input

The environment consists of several independent objects and surrounding walls. A viewer is located in the environment that is encoded by a set of views fixating on each object. When the viewer locates at \mathbf{x} and fixates on an object, O_i of which location is $\mathbf{d}_i - \mathbf{x}$, then a view, $S_{\mathbf{d}_i}(\phi)$, is given by

$$S_{\mathbf{d}_i}(\phi) = \begin{cases} P(\phi - \theta_{\mathbf{d}_i}) & (-C^F < \phi < C^F) \\ \emptyset & (\text{otherwise}) \end{cases}, \quad (1)$$

where ϕ denotes eccentricity, $\theta_{\mathbf{d}_i}$ denotes an orientation to the object from the viewer, $P(\phi)$ denotes a panoramic view at the viewer location \mathbf{x} , and C^F denotes a size of visual field.

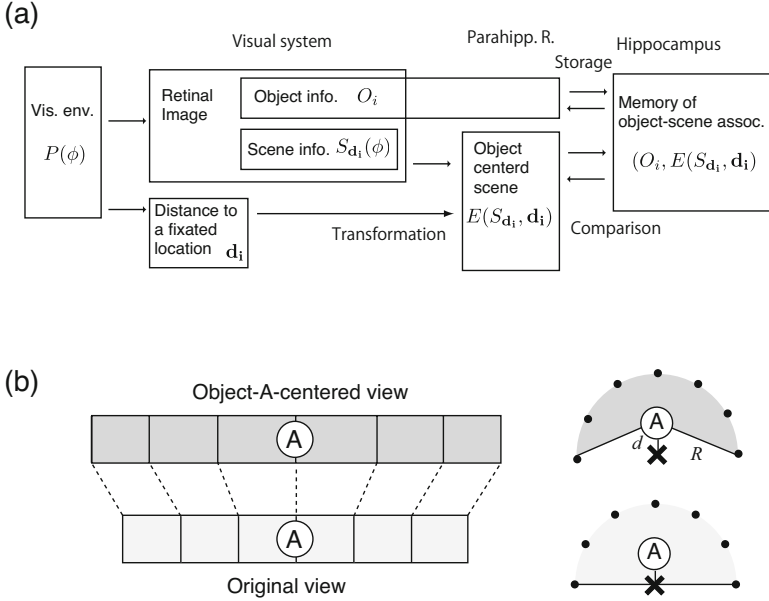


Fig. 2. (a) The model consisting of the visual system, and the hippocampus. (b) Viewer's view and object-centered view under an equal distance assumption. Right figure shows arrangement of viewer (cross), object (circle) and equally distant landmarks (dots). Displacements of landmarks are simply given by $\frac{d}{R} \sin(\phi)$.

2.2 Transformation of the Viewer Scene into the Object-Centered Scene

The viewer's view is memorized as a combination between object and scene (a background gray-scale luminance pattern) information where the scene is perspectively translated to include information of object location itself. During encoding, the scene is 'expanded' according to distance to fixation location by using an equal distance assumption [13] where distances to portions of wall is assumed to be constant (Fig. 2b). When distances to object and wall portions are respectively given by d and R , displacement of wall portions are simply described by $\frac{d}{R} \sin(\phi)$. This assumption works perfectly when the walls are sufficiently distant from the object, and it is also shown to be available for navigation in real environment [13]. Importantly, this translation is associated with optical flow field caused by self-motion that is well investigated as a neuronal selectivity in MST region, thus it could be expected to be biologically plausible. An expected object-centered scene, $E(\phi | S, \mathbf{d})$, is given by

$$E(\phi | S, \mathbf{d}) = S(\phi - \theta_{\mathbf{d}} - \frac{|\mathbf{d}|}{R} \sin(\phi)), \quad R \gg |\mathbf{d}| \quad (2)$$

where \mathbf{d} a fixation location from the viewer. Note that this translation is nonlinear, i.e., $E(S, \mathbf{d}_1) \neq E(E(S, \mathbf{d}_2), \mathbf{d}_3)$ under $\mathbf{d}_1 = \mathbf{d}_2 + \mathbf{d}_3$. This is geometrically inaccurate, while it is expected to be available for representing object's location in the environment as a neuronal representation.

2.3 Memory Storage

As demonstrated in our previous reports [10], multiple object–scene associations are successively encoded and assumed to be stored in the CA3 associative network in the hippocampus. When the environment includes n objects, n object–scene associations will be stored.

2.4 Estimation of Displacement between Scenes

During retrieval and imagery, a displacement of object pair is calculated as a comparison of two scenes by using the translation shown in eq.(2). When two scenes, $S_{\mathbf{a}}$ and $S_{\mathbf{b}}$, are given, their expected displacement vector, \mathbf{q} , is given by

$$(\mathbf{q}(S_{\mathbf{a}}, S_{\mathbf{b}}), \varphi) : \sum_{\phi=0}^{2\pi} (S_{\mathbf{a}}(\phi - \varphi) - E(\phi|S_{\mathbf{b}}(\phi - \theta_{\mathbf{q}}), \mathbf{q}))^2 \rightarrow \min. \quad (3)$$

2.5 Experimental Procedure

Ability of mental imagery of object arrangement at voluntary location is tested by a square environment consisting of three objects and walls having gradually changing luminance in space (Fig. 3A). Viewer is fixed at (10, 70) and object locations are (70, 120), (90, 80) and (120, 130). The mental imagery is defined by a comparison of an imagery scene translated for a fixation location, \mathbf{r} , and a memorized object–centered scene. By this way, orientation and distance of each object at the imagery location \mathbf{r} is obtained. The ability of mental imagery is evaluated by a difference between imagery and geometrical objects' distance and orientation. The error in the imagery of i -th object orientation in comparison with a real orientation is given by

$$\mathbf{e}_{\mathbf{r}}^{\text{I}}(i) = (\mathbf{d}_i - \mathbf{r}) - \mathbf{q}(E(\phi|S_{\mathbf{d}_i}, \mathbf{d}_i), E(\phi'|S_{\mathbf{r}}, \mathbf{r})). \quad (4)$$

where \mathbf{d}_i is displacement of i -th object and \mathbf{r} denotes imagery location from the viewer location.

Furthermore, to evaluate perspective accuracy of object–centered scenes, memorized object–centered scenes are compared with real scene at location \mathbf{r} . When the object–centered scene is perspectiveally correct, the resultant distance and orientation will be identical with real object arrangement, otherwise the object–centered scene is considered to be perspectiveally inaccurate, i.e., these might include spatial information but not correspond with real scenes. The displacement error of i -th object between recalled and real view is given by

$$\mathbf{e}_{\mathbf{r}}^{\text{R}}(i) = (\mathbf{d}_i - \mathbf{r}) - \mathbf{q}(E(\phi|S_{\mathbf{d}_i}, \mathbf{d}_i), P_{\mathbf{r}}). \quad (5)$$

In the results, a mean direction error, $\sum_i \theta_{e_r^I(i)}/3$, and a mean distance error at each imagery viewpoints, $\sum_i |e_r^I(i)|/3$ were calculated for every novel locations.

3 Results

3.1 Object Arrangement and Scene Estimation at the Imagery Viewpoint

Figure 3 shows a result of an expected view at an imagery viewpoint (80,30). During encoding, three object–scene association are memorized, where object–centered scenes are generated by translating original viewer’s view. Each view is shown as a shaded ring plot to represent their orientational relationship to the environment. Mental imagery is given by a comparison between an imaginary scene fixated at (80,30) and memorized object–centered scenes. By calculating optimal transformation for matching to object–centered view (eq.(3)), displacement to each object at the imagery location is obtained. In Fig.3, these displacement are plotted as lines from the imagery location. It is shown that the displacement errors are small and object arrangement is correctly reconstructed.

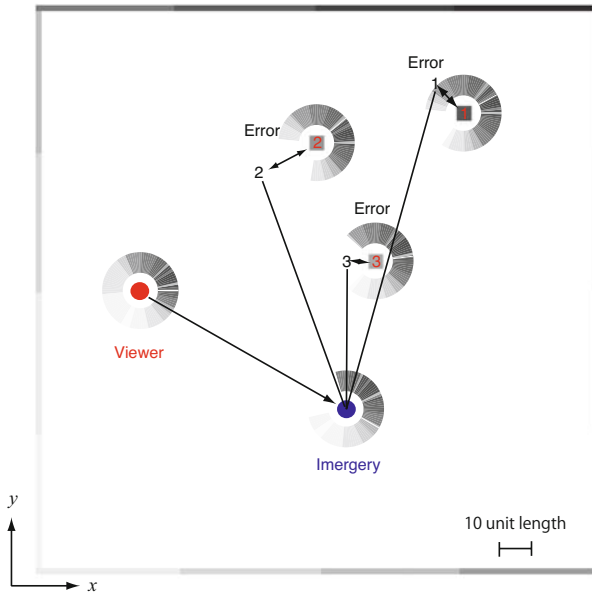


Fig. 3. The result of an object arrangement estimation at an imagery viewpoint. The red circle and numbers indicate the viewer and objects locations, respectively. The circle and numbers in blue indicate the imagery viewer and estimated object locations, respectively. The shaded ring plot shows the viewer’s panoramic view and the fan-shaped plots represent expected scenes at each location.

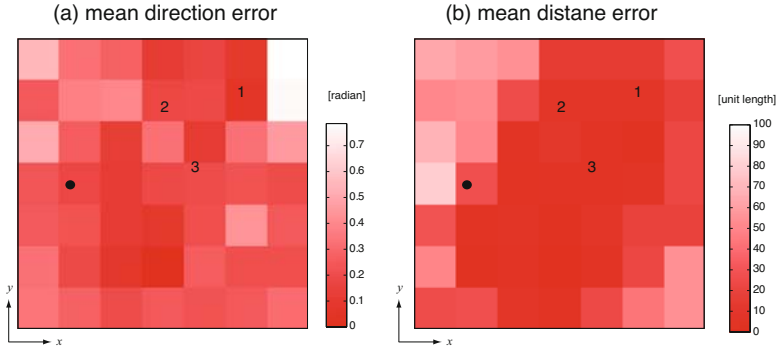


Fig. 4. Errors in the object arrangement estimation at each imagery viewpoint. Large squares indicate the square arena and the viewer and objects locations are indicated by a dot and numbers, respectively. (a) Mean direction errors and (b) mean distance errors at each imagery viewpoint are shown.

To evaluate the ability to generate object arrangement at imagery location, we calculated the errors with changing imagery location with a spacing of 20 unit length. Figure 4 shows a result of errors in the imagery object arrangement. In a wide area, direction errors are less than 0.5 radian and distance errors are less than 50 unit length. At locations close to walls, the error increases, while the distance to the wall does not appear critical for calculating imagery object arrangement. These results indicate that the object-centered scene representation under the equal distance assumption can produce the ability to calculate imagery object arrangement.

3.2 Retrieval of Object Arrangement at a Novel Viewpoint

In this section, a perspective accuracy of object-centered scene representation was evaluated by using a real scene at the novel location for the object arrangement estimation. Figure 5 shows a result of errors in the retrieval of the object arrangement. The errors in the object direction estimation (Fig. 5a) are large around the objects' locations. This is reasoned by that the neighbor objects widely occlude the background scene and disturb the object displacement estimation in eq.(3). On the other hand, the errors in the object distance estimation (Fig. 5b) are large near the walls. It could be concluded that the absolute distance information is difficult to be conveyed by the object-centered scene under the equal distance assumption in eq.(2).

4 Discussions

The object-centered scene representation was proposed as a component of the environmental memory in the hippocampus. The computational abilities of the representation for the spatial imagery (Section 3.1) and the retrieval (Section 3.2)

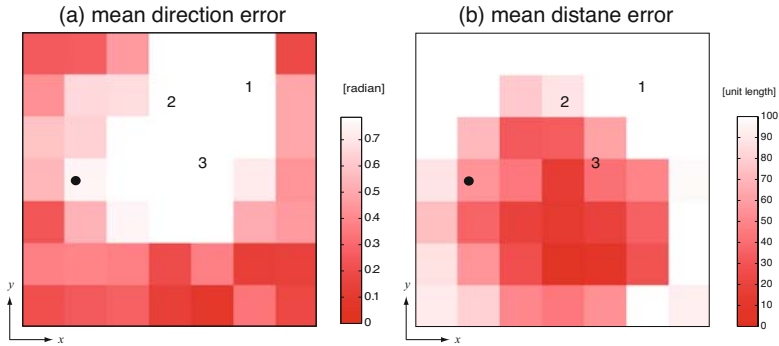


Fig. 5. Errors in the memory retrieval of object arrangement estimation by a real scene at a novel viewpoint. (a) Mean direction error and (b) Mean distance error at each novel viewpoints are shown.

were evaluated by using an object arrangement estimation error at novel viewpoint. The results demonstrate that the memory of object and object-centered scene associations at a viewpoint could be available for generating a new object arrangement at a novel viewpoint. It is shown that the object-centered scene representation could play a fundamental role for mediating the egocentric space to the allocentric space (Fig. 6). Such a memory representation has a similarity to a conceptual model with fragments memory [15], while our model has advantage in the calculation of orientation and distance between voluntary object pair.

Neurophysiological studies have demonstrated that the object-centered space representation exists in the parietal region [16] [17]. More importantly the parietal region has massive afferent connections to the hippocampus through the parahippocampal region [18] which is known to represent environmental scene information [19]. The object-centered scene representation used in our model would be expected to be represented by the parahippocampal cortex.

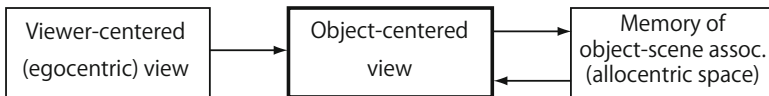


Fig. 6. Transformation of spatial information proposed by our computational model

Acknowledgments. This study was supported by the MEXT KAKENHI (20220003).

References

1. O’Keefe, J., Nadel, L.: The Hippocampus as a Cognitive Map. Clarendon Press, Oxford (1978)
2. Muller, R.U., Kubie, J.L., Saypoff, R.: The Hippocampus as a Cognitive Graph (abridged version). *Hippocampus* 1(3), 243–246 (1991)

3. Samsonovich, A., McNaughton, B.L.: Path Integration and Cognitive Mapping in a Continuous Attractor Neural Network Model. *J. Neurosci.* 17(15), 5900–5920 (1997)
4. Barry, C., Burgess, N.: Learning in a Geometric Model of Place Cell Firing. *Hippocampus* 17, 786–800 (2007)
5. Hafting, T., Fyhn, M., Molden, S., Moser, M.B., Moser, E.I.: Microstructure of a Spatial Map in the Entorhinal Cortex. *Nature* 436(7052), 801–806 (2005)
6. Smith, M.L., Milner, B.: The Role of the Right Hippocampus in the Recall of Spatial Location. *Neuropsychologia* 19, 781–793 (1981)
7. King, J.A., Burgess, N., Hartley, T., Vargha-Khadem, F., O’Keefe, J.: Human Hippocampus and Viewpoint Dependence in Spatial Memory. *Hippocampus* 12, 811–820 (2002)
8. Stepankova, K., Fenton, A.A., Pastalkova, E., Kalina, M., Bohbot, V.D.: Object-location Impairment in Patient with Thermal Lesions to the Right or Left Hippocampus. *Neuropsychologia* 42, 1017–1028 (2004)
9. Rolls, E.T., Stringer, S.M., Trappenberg, T.P.: A Unified Model of Spatial and Episodic Memory. *Proc. R. Soc. Lond. B* 269, 1087–1093 (2002)
10. Sato, N., Yamaguchi, Y.: On-line Formation of a Hierarchical Cognitive Map for Object–Place Association by Theta Phase Coding. *Hippocampus* 15, 963–978 (2005)
11. Byrne, P., Becker, S., Burgess, N.: Remembering the Past and Imagining the Future: A Neural Model of Spatial Memory and Imagery. *Psychol. Rev.* 114(2), 340–375 (2007)
12. Hartley, T., Burgess, N., Lever, C., Cacucci, F., O’Keefe, J.: Modeling place fields in terms of the cortical inputs to the hippocampus. *Hippocampus* 10(4), 369–379 (2000)
13. Franz, M.O., Schölkopf, B., Mallot, H.A., Bühlhoff, H.H.: Where Did I Take that Snapshot? Scene-based Homing by Image Matching. *Biol. Cybern.* 79, 191–202 (1998)
14. Hartley, T., Bird, C.M., Chan, D., Cipolotti, L., Husain, M., Vargha-Khadem, F., Burgess, N.: The Hippocampus Is Required for Short-term Topographical Memory in Humans. *Hippocampus* 17(1), 34–48 (2007)
15. Worden, R.: Navigation by Fragment Fitting: A Theory of Hippocampal Function. *Hippocampus* 2(2), 165–188 (1992)
16. Olson, C.R.: Object-based Vision and Attention in Primates. *Curr. Opin. in Neurobiol.* 11, 171–179 (2001)
17. Crowe, D.A., Averbeck, B.B., Chafee, M.V.: Neural Ensemble Decoding Reveals a Correlate of Viewer–to Object-Centered Spatial Transformation in Monkey Parietal Cortex. *J. Neurosci.* 28(20), 5218–5228 (2008)
18. Suzuki, W.A., Amaral, D.G.: Perirhinal and Pparahippocampal Cortices of the Macaque Monkey: Cortical Afferents. *J. Comp. Neurol.* 350(4), 497–533 (1994)
19. Epstein, R.A.: Parahippocampal and Retrosplenial Contributions to Human Spatial Navigation. *Trends Cogn. Sci.* 12(10), 388–396 (2008)

Biophysical Modeling of a *Drosophila* Photoreceptor

Zhuoyi Song^{1,2}, Daniel Coca¹, Stephen Billings¹, Marten Postma³,
Roger C. Hardie³, and Mikko Juusola²

¹ Department of Automatic Control and Systems Engineering, University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK

² Department of Biomedical Science, University of Sheffield, Western Bank, S10 2TN, Sheffield, UK

³ Department of Physiology, Development and Neuroscience, Cambridge University, Downing Street, Cambridge, CB2 3DY, UK
{zhuoyi.song,d.coca,S.Billings}@sheffield.ac.uk,
m.postma@uva.nl, rch14@cam.ac.uk, m.juusola@sheffield.ac.uk
<http://www.shef.ac.uk/acse/spcs>

Abstract. It remains unclear how visual information is co-processed by different layers of neurons in the retina. In particular, relatively little is known how retina translates vast environmental light changes into neural responses of limited range. We began examining this question in a bottom-up way in a relatively simple fly eye. To gain understanding of how complex bio-molecular interactions govern the conversion of light input into voltage output (phototransduction), we are building a biophysical model of the *Drosophila* R1-R6 photoreceptor. Our model, which relates molecular dynamics of the underlying biochemical reactions to external light input, attempts to capture the molecular dynamics of phototransduction gain control in a quantitative way.

Keywords: Biophysical model, *Drosophila* photoreceptor, phototransduction cascade, Gillespie algorithm, Hodgkin-Huxley model.

1 Introduction

There have been many approaches to model fly photoreceptors [17,15,14,13]. van Hateren produced a linear-nonlinear cascade model to compare phototransduction in blowfly photoreceptors to that of primate cones [17]; Pumir and his co-workers produced a biophysical model of fly phototransduction cascade [15]; Váhásórinki et al. developed a Hodgkin-Huxley model, which relates Light Induced Current (*LIC*) to voltage response, to study the effect of voltage-gated potassium channels on visual information processing [10]. There are also models for intracellular calcium dynamics, such as the diffusion model introduced by Postma et al. [14] and the calcium homeostasis model by Oberwinkler [11].

To begin to investigate how a network of photoreceptors and interneurons, whose responses are shaped together through feed-forward and feedback synapses,

co-process visual information, we developed a new biophysical model for *Drosophila* photoreceptor, which will form the input stage for a more complex network model that will be developed in the near future. Our model describes both photo-sensitive and photo-insensitive membranes of the photoreceptor. The photo-sensitive part of the model consists of linear and nonlinear differential equations describing biochemical reactions involved in phototransduction cascade. The photo-insensitive membrane is represented by an electrical circuit model based on Hodgkin-Huxley formalism. The complete model can predict quite well macroscopic current and voltage responses to varying light impulses (patch-clamp data from whole cell recordings).

2 Structure of *Drosophila* Photoreceptor

The compound eye of *Drosophila* (Fig. 1A) contains 776 ommatidia, stereotypical processing units that focus the light energy by a corneal lens onto the rhabdom, the light-sensitive parts of the photoreceptors underneath. Inside of each ommatidium, the outer photoreceptors (R1-R6) are arranged in a ring, surrounding the inner R7 and R8 photoreceptors, which are stacked on top of each other in the center. This gives ommatidia a characteristic pattern of 7 disks when viewed from the top or in cross-section (Fig. 1D). R1-R8 are arranged around a central space, intraommatidial cavity. Fig. 1E shows that *Drosophila* photoreceptors are thin elongated cells, $100\ \mu\text{m}$ in length (excluding axon) and $5 - 6\ \mu\text{m}$ in diameter. Their plasma membranes divide into photo-sensitive (rhabdomere) and

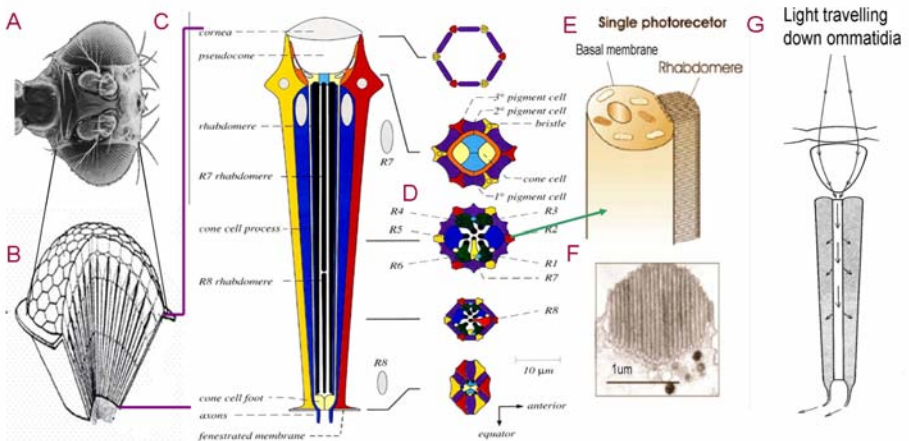


Fig. 1. Anatomy of *Drosophila* eye. (A) The head. (B) Slice of a compound eye. (C) Vertical section of ommatidium. (D) Cross section of ommatidium. (E) Schematic single photoreceptor. (F) Cross section of Rhabdomere. (G) Light pathway. (C) and (D) are modified from [18]. (E) and (F) are modified from [3]. (G) is from D. G. Mackean (<http://www.biology-resources.com/drawing-ommatidium-refraction.html>)

photo-insensitive membrane (basal membrane). The rhabdomere transduce light into current (LIC), while the basal membrane incorporates different species of voltage-gated K^+ channels, which help to convert LIC into a well-defined voltage response. Rhabdomere (cross-section shown in Fig. 1F) consist of 30,000 finger-like protrusions (microvilli) into the central space. Each microvillus in a rhabdomere is believed to act independently as a phototransduction unit, capturing photons and transducing light energy to a current, which is then used to charge the plasma membrane to generate a voltage response (Fig. 1G).

3 The Model of Photoreceptor

3.1 Photoreceptor Model Structure

The proposed photoreceptor model can be decomposed to several modules, as shown in Fig. 2. The first module (Fig. 2A) corresponds to a random photon capture model, which accounts for the fact that the number of photons absorbed by each microvillus varies across the rhabdomere. The input to this module is a 1 ms light impulse and the output represents the number of photons absorbed by each microvillus. To prevent lateral interactions between microvilli and to keep the integration of LIC linear, the light input was given the maximum effective brightness of 1,000 absorbed photons (1,000,000 photons/s). For this brief stimulation, all photons are assumed to be absorbed at the same time instant. The randomness of photon capture is based on Poisson statistics [4]. It is important to note that $LIC/photon$ (average light induced current per photon) produced in an individual microvillus changes with the number of photons it absorbs. Consequently, it is crucial to have a random photon capture model to produce the light input for each microvillus.

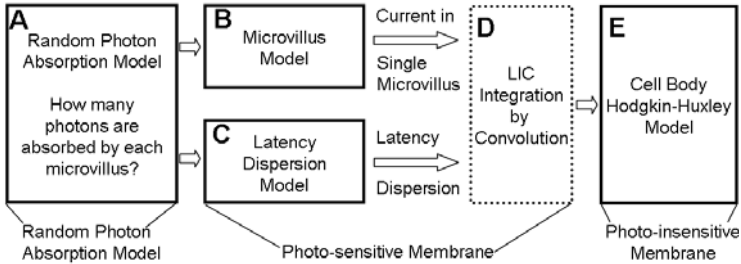


Fig. 2. Schematic structure of our model for impulse light response of *Drosophila* photoreceptor. (A) Random photon absorption model. (B) Deterministic model for phototransduction cascade. (C) Stochastic model for latency dispersion. (D) LIC integration by convolution to produce macroscopic current. (E) Hodgkin-Huxley model for the cell body.

Similar to the anatomical division of the photoreceptor membrane, the processing of light stimuli is performed in two stages. The first processing stage, implemented in modules in Fig. 2B, C, and D, produces the macroscopic LIC

from rhabdomere (photo-sensitive membrane). These signals then drive the second processing stage, a model of the photo-insensitive membrane implemented in Fig. 2E, which accounts for the dynamics of the known voltage-gated ion-channels on the cell body. The processing within a rhabdomere is divided into two parts. The first part (Fig. 2B) is a deterministic model for biochemical reactions of phototransduction cascade within a single microvillus, based on coupled differential equations. The second part (Fig. 2C) is a latency dispersion model that accounts for variations in signal transduction between different microvilli. The latency distribution is obtained through stochastic simulation (Gillespie algorithm) of the phototransduction model. The macroscopic current injected to the cell body is obtained from integration of *LIC* produced in all microvilli. Under our linear current integration assumption, the integration is produced by convolving the basic current bump (generated by deterministic phototransduction model) with the latency dispersion (Fig. 2D) [19,5].

3.2 Random Photon Absorbtion Model

The random photon absorbtion model is characterized in terms of the following parameters: N_{micro} : the number of microvilli in the whole rhabdomere; N_m : the number of activated microvilli; N_{photon} : the number of photons for the light impulse; $N_p(m_j)$: the number of photons captured by each activated microvillus m_j , $m_j = 1, 2, \dots, N_m$; λ_M : The average number of light quanta absorbed per microvillus; f_x : the fractions of microvilli that absorb $x = 0, 1, 2 \dots$ light quanta; f_e : the fraction of microvilli that escape photo-activation; f_a : the fraction of light activated microvilli; λ_p : the average number of photons absorbed by each activated microvillus; $p(k)$: the selection possibility to absorb k photons for each microvillus; k_m : the maximum number of photons each microvillus could absorb; $q(k)$: the accumulation photon selection probability.

The calculation contains two steps. First, N_m is calculated iteratively.

1. Initialization. N_{photon} ($N_{photon} < 1000$), $N_{micro} = 30,000$, $N_m = N_{micro}$ (N_m is initially set to N_{micro} , assuming all microvilli are activated).
2. Calculate $\lambda_M = \frac{N_{photon}}{N_m}$.
3. Assuming that f_x follow a Poisson distribution: $f_x = \frac{e^{-\lambda_M} * \lambda_M^x}{x!}$. Therefore, $f_e = e^{-\lambda_M}$ and $f_a = 1 - e^{-\lambda_M}$.
4. Update N_m and return to 2 until N_m converged (the termination criteria is heuristic, here, $N_m(i+1) - N_m(i) < 10$, i is the index of current iteration loop).

Then $N_p(m_j)$ is determined based on Poisson distributed roulette rule.

1. Compute λ_p as $\lambda_p = \frac{N_{photon}}{N_m}$.
2. The probability that an activated microvillus m_j can absorb k photons, assuming Poisson distribution, is given by $p(k) = \frac{e^{-\lambda_p} * \lambda_p^k}{k!}$. Here, because $N_{photon} \ll N_{micro}$, we assume that $p(k) = 0$ if $k > k_m$, where $k_m = 10 * \text{round}(\lambda_p + 1)$ ($\text{round}(x)$ obtains the nearest integer of x).

3. Compute $q(k) = \frac{\sum_{j=1}^k p(j)}{\sum_{j=1}^{k_m} p(j)}$, generate a random number r , if $q(k) < r < q(k+1)$, $N_p(m_j) = k$.

Fig. 3 shows simulation results of random photon absorption model for a light impulse that contains 600 photons. The number in the x-axis is the number of 'activated microvilli', which is quoted because some of the 'activated microvilli' might absorb 0 photons, meaning failures. The y-axis is the number of photons absorbed by each microvillus. Then microvilli are grouped into different categories based on the number of photons they absorbed ($C(P_h)$ stores the number of microvilli that absorb P_h photons), as the signal transduction properties ($LIC/photon$) vary with this number ($P_h = 1, 2, \dots, \max(N_p)$).

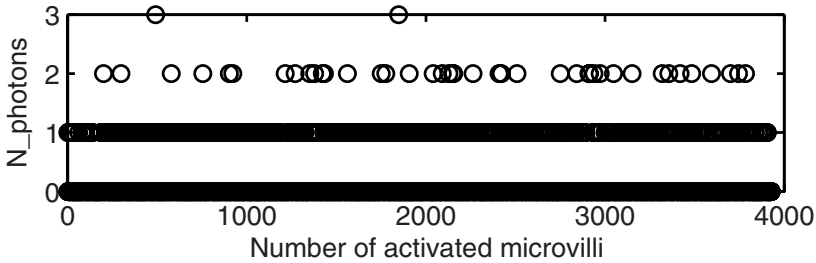


Fig. 3. Simulation of random photon absorption model

3.3 Model for Phototransduction Cascade

Molecularbiology of Phototransduction cascade. Although the phototransduction cascade is not fully characterized, it is clear that the photopigment - rhodopsin, thousands of which are densely packed on the microvillar membrane - will change its conformation upon absorption of a photon. This activated rhodopsin (metarhodopsin) then activates a second messenger, G -protein. While bound to metarhodopsin (M), G -protein exchanges inactive guanosine diphosphate (GDP) for active guanosine triphosphate (GTP), which in turn catalyzes phospholipase C (PLC). G -protein coupled PLC cleaves phosphatidyl 4,5-bisphosphate (PIP_2) into two intracellular messengers: inositol trisphosphate (IP_3) and diacylglycerol (DAG). IP_3 is soluble in the cytosol, while DAG is insoluble and remains bounded to the membrane of microvilli. It is believed that DAG , or its metabolite Polyunsaturated Fatty Acids ($PUFA$), are the excitation messengers to the cation selective ion channels $TRP/TRPL$. The opening of these transduction-channels fluxes in permeable ions, Na^+ , Ca^{2+} , Mg^{2+} , generating LIC inside a single microvillus (for review, see [3]). Fig. 4 shows a simplified diagram for *Drosophila* phototransduction cascade.

Regulation of *Drosophila* phototransduction cascade. Molecular, genetic, and physiological studies suggest that at least 20 different gene products are dedicated to the functioning and regulation of this one signaling cascade

in *Drosophila* [3]. There are positive feedback pathways to speed up excitation. *TRP* channels have a 'all-or-none' excitation property, arising from Ca^{2+} dependent positive feedback to *TRP* channels. When the first *TRP* channel opens, the fluxed in Ca^{2+} will excite other *TRP* channels inside microvillus, triggering many *TRP* channels to open, until free intracellular calcium ($[Ca^{2+}]_i$) inside microvillus build up to a level that terminates responses. In addition to excitation, photoreceptor neurons have evolved sophisticated mechanisms for quick termination of *LIC* (deactivation) to maintain sensitivity. In *LIC* termination, Ca^{2+} and calmodulin (*CAM*, Ca^{2+} binding protein, acting as a Ca^{2+} buffer in cytosol) play important roles as negative feedback signals, acting on many target molecules in the phototransduction cascade [2]. Not only can Ca^{2+} provide negative feedback signals to *TRP*, *TRPL* channels to facilitate the closure of the channels, but it can also reduce *PLC* activity, facilitate the binding of arrestin to metarhodopsin (the inactivation process of meta-rhodopsin) [7], *etc.*

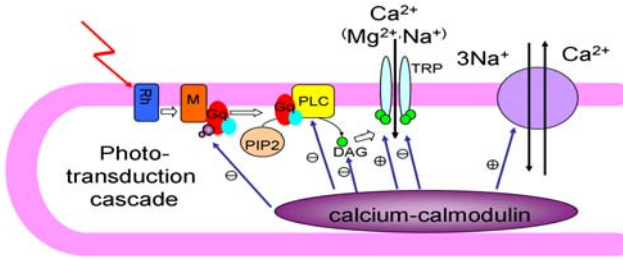


Fig. 4. Phototransduction cascade illustration

3.4 Mathematical Description of Phototransduction Model: Kinetic Equations

The phototransduction cascade model was modified from [15]. The main difference between the models is in Ca^{2+} homeostasis (Eq. 9 to Eq. 12 vs. Eq. 7 and 8 in [15]). The balances, or dynamics, in production and consumption of vital molecules are modeled by nonlinear first-order differential equations. For some of the molecules that are in small numbers, the units are counts of molecules, otherwise, we use concentration (the two are related by the microvillus volume factor, $3 \times 10^{-12} \mu l$). To ignore noise effects, all variables are calculated as expectations. In the following equations, the notation X denotes the expected number of molecules, and X^* will refer to the active state of X , whereas $[X]$ denotes concentration, $[X]_i$ is for intracellular concentration and $[X]_o$ for extracellular concentration. Rates of activation are generically denoted as κ and rates of deactivation denoted as γ .

$$\frac{dM^*}{dt} = -\gamma_{M^*} \times (1 + g_{M^*} f_n) \times [M^*]. \quad (1)$$

Eq. 1 (vs. Eq. 1 in [15]) is for Metarhodopsin (M). Since all photons are assumed to be effectively absorbed at $t = 0$, there is one-to-one mapping between

number of photons and the value of M . Hence, M is initialized as $M^*(0) = P_h$. This equation describes the decay of M^* . Compared to [15], we have introduced an additional operator ($\lceil \cdot \rceil$) to avoid negative and non-integer numbers of metarhodopsin. The notation $\lceil M^* \rceil$ means the smallest integer that is bigger than M^* if $M^* > 0$, otherwise $\lceil M^* \rceil = 0$. The f_n term on the right-hand-side (defined in Eq. 8) is negative feedback from C^* (Ca^{2+} bound CAM). This term is introduced to represent the facilitation of M^* inactivation by C^* .

$$\frac{dG}{dt} = -\kappa_{G^*} \times G \times \lceil M^* \rceil + \gamma_G \times (G_T - G - G^*) + \kappa_{PLC^*} \times PLC^* \times G^*. \quad (2)$$

$$\frac{dG^*}{dt} = \kappa_{G^*} \times G \times \lceil M^* \rceil - \kappa_{PLC^*} \times PLC_T \times G^*. \quad (3)$$

Eqs. 2 and 3 describe activation of G -protein by M^* . There are three states of G -protein, G_qGDP is denoted by G and G^* represents G_qGTP (active state of G -protein), while the nucleotide-free state of G -protein is calculated as $G_T - G - G^*$ (G_T is the total number of G -protein inside one microvilli). The first terms in Eq. 2 and in Eq. 3 are modeling exchange from GDP to GTP of G , stimulated by M^* . The second term in Eq. 2 is for stabilizing nucleotide-free state G -protein by GDP . The third term in Eq. 2 is added on to Eq. 2 in [15] to model the formation of G upon deactivation of G^* by $GTPase$ activity stimulated by PLC^* . The second term in Eq. 3 has two roles in forming the profile of G^* . One role is the conversion of G^* to PLC complex (PLC^*) by binding to PLC ($\kappa_{PLC^*} \times (PLC_T - PLC^*) \times G^*$, the same with the first term in Eq. 4) and the other role is the conversion of G_qGTP to G_qGDP by PLC^* ($\kappa_{PLC^*} \times PLC^* \times G^*$, the last term in Eq. 2).

$$\frac{dPLC^*}{dt} = \kappa_{PLC^*} \times (PLC_T - PLC^*) \times G^* - \gamma_{PLC^*} \times (1 + g_{PLC^*} f_n) \times PLC^*. \quad (4)$$

Eq. 4 represents the dynamics of PLC^* , active PLC complex formed by G^* and PLC . The last term in Eq. 4 describes deactivation of PLC^* , which was also assumed to be accelerated by negative nonlinear feedback from C^* .

$$\frac{dD^*}{dt} = \kappa_{D^*} \times PLC^* - \gamma_{D^*} \times (1 + g_{D^*} f_n) \times D^*. \quad (5)$$

PLC^* then cleaves PIP_2 into DAG and IP_3 . There is a recycling pathway for PIP_2 , but it is much slower than a bump generation ($\sim 1,000$ times slower, calculated from time constants of the two processes [3]). Hence the dynamics of this recycling is omitted here, leading to a proportional relationship between PIP_2 consumption to number of PLC^* . The response property of second messenger (presumably DAG) could be related directly to PLC^* and is described by Eq. 5. The interpretation of this equation would be the dynamical balance between the production of DAG from PIP_2 and its degradation through action of DAG -kinase.

$$\frac{dT^*}{dt} = \kappa_{T^*} \times (1 + g_{T^*,p} f_p) \times \left(\frac{D^*}{K_{D^*}}\right)^m \times (T_T - T^*) - \gamma_{T^*} \times (1 + g_{T^*,n} f_n) \times T^*. \quad (6)$$

Eq. 6 describes opening of *TRP* and *TRPL* channels (as in 15, we use one equation to describe these two types of channels for simplicity), with T^* denoting the number of open state channels and T_T the total number of channels, which is conserved inside one microvillus. The precise mechanism of *TRP/TRPL* activation is not known, but it is likely that 2nd messenger molecules (e.g. DAG) act cooperatively to open one channel. Hence, in Eq. 6, the activation rate of T^* is in proportion to $(\frac{D^*}{K_{D^*}})^m$, where m is the cooperativity parameter for *DAG* molecules and is set to be 4 here).

$$f_p([Ca^{2+}]_i) = \frac{([Ca^{2+}]_i/K_p)^{m_p}}{1 + ([Ca^{2+}]_i/K_p)^{m_p}}. \quad (7)$$

In the dynamics of activation of *TRP/TRPL* channels, positive feedback signal from Ca^{2+} is included because of the 'all or none' activation properties of these channels. This feedback is formulated as a Hill function of $[Ca^{2+}]_i$ inside microvillus (Eq. 7), where K_p is the dissociation constant, which is $[Ca^{2+}]_i$ that provide half occupancy of Ca^{2+} binding sites for the channels. m_p is the Hill coefficient, describing the cooperativity of Ca^{2+} in exciting the channels. For the acceleration of *TRP/TRPL* deactivation (refractory transition from open to closed state of the channels), negative feedback is also provided from C^* , the same as the negative feedbacks to other signalling components in the cascade (M^* , PLC^* , D^* , etc). This negative feedback is a sigmoidal shaped function of C^* :

$$f_n([C^*]) = \frac{([C^*]/K_n)^{m_n}}{1 + ([C^*]/K_n)^{m_n}}. \quad (8)$$

where K_n is the dissociation constant and m_n Hill coefficient for C^* . In reality, the affinity of C^* might vary for different feedback targets, leading to different values of parameters K_n and m_n . However, for simplicity, we look at the whole pool of available C^* binding sites as the same affinity properties. Feedback strengths are parameterized by g_i . This simplification provides a practical initial approximation, in absence of more complete mechanistic knowledge about the different underlying processes.

The spontaneous activities of all the molecules in the dark, which act as a noise source for the real system, are ignored. Hence, the initial values for the differential equations (Eq. 1 to Eq. 6) are set as $G(0) = 50$, $G^*(0) = 0$, $PLC^*(0) = 0$, $D^*(0) = 0$, $T^*(0) = 0$.

The dynamics of $[Ca^{2+}]_i$ are of particular interests since $[Ca^{2+}]_i$ serves as feedback signal to many targets in the phototransduction cascade. The driving force for $[Ca^{2+}]_i$ is Ca^{2+} entry through *TRP/TRPL* channels during light response. This Ca^{2+} influx (I_{Ca}) into a microvillus is modeled by Eq. 9:

$$I_{Ca} = P_{Ca} \times I_{T^*} \times T^*. \quad (9)$$

I_{T^*} is the average current fluxed into the cell per *TRP* channel (~ 0.68 pA/*TRP*) and P_{Ca} ($\sim 40\%$) represents the percentage of Ca^{2+} out of the total current influx (~ 10 pA). At peak response, the Ca^{2+} influx is as high as 10^7 ions/s.

Owing to the small volume of a single microvillus, local $[Ca^{2+}]_i$ can rise dramatically. It could peak, for example, at 100 mM during a 20 ms quantum bump, if no other processes were counterbalanced with the influx. In comparison, $[Ca^{2+}]_i$ is about 0.16 μM in the dark state. However, it is important to maintain $[Ca^{2+}]_i$ homeostasis because Ca^{2+} is toxic to the cell in high concentrations.

Apart from Ca^{2+} entry, we model three other processes that modulate $[Ca^{2+}]_i$ dynamics: (i) Ca^{2+} extrusion through Na^+/Ca^{2+} exchanger; (ii) Ca^{2+} buffering by calmodulin; (iii) Ca^{2+} diffusion to the cell body. Na^+/Ca^{2+} exchanger is a conventional transport system with a stoichiometry 3:1, *i.e.* 3 Na^+ ions are exchanged for 1 Ca^{2+} ion. This ratio results in a net charge imbalance, which produces a weakly depolarizing current. The Ca^{2+} current, extruded by the exchanger, is two times the net exchanger current. The net Ca^{2+} influx is obtained by subtracting Ca^{2+} extrusion (through Na^+/Ca^{2+} exchanger) from total Ca^{2+} influx (through *TRP* channels): $I_{Ca,net} = I_{Ca} - 2 \times I_{NaCa}$, where I_{NaCa} denotes net inward current through Na^+/Ca^{2+} exchanger. The formulation for Na^+/Ca^{2+} exchanger current is adapted from Luo-Rudy model for cardiac cells [8] and is comparable to other models for cardiac myocyte [16]. The model is derived based on thermodynamics of electro-diffusion [9], which assume that the sole source of energy for Ca^{2+} transport is the Na^+ electrochemical gradient.

$$I_{NaCa} = K_{NaCa} \times \frac{1}{K_m, Na^3 + [Na]_o^3} \times \frac{1}{K_m, Ca + [Ca]_o} \times \frac{\exp(\eta \frac{VF}{RT}) [Na]_i^3 [Ca]_o - \exp((\eta-1) \frac{VF}{RT}) [Na]_o^3 [Ca]_i}{1 + d_{NaCa} \exp((\eta-1) \frac{VF}{RT})} \quad (10)$$

where K_{NaCa} , d_{NaCa} are scaling factors, η denotes the (inside) fractional distance into the membrane of the limiting energy barrier. V is the transmembrane potential in volts, ideally this should be from the membrane potential of the cell body. However, as in the simulation, the membrane potential is generated off-line by a separate cell body model, this was approximated by the membrane potential generated by a single Quantum bump. F is the Faraday constant, (96,485 $C \times mol^{-1}$). R is the gas constant (8.314 $J \times K^{-1} \times mol^{-1}$) and T is the absolute temperature, measured in kelvins.

Another Ca^{2+} extruding option might be through the Ca^{2+} uptake by buffering proteins, such as *CAM* (0.5 mM), which are abundant inside microvillus. The diffusion of buffer molecules over the time scale of interest could be omitted because of the relatively large molecular weight. This binding dynamic was modeled as a first-order process [16]:

$$\frac{dO_c}{dt} = K_U [Ca^{2+}]_i (1 - O_c) - K_R O_c. \quad (11)$$

where, O_c is the buffer occupancy, *i.e.* the fraction of sites already occupied by Ca^{2+} ions, and therefore unavailable for Ca^{2+} binding. $\frac{dO_c}{dt}$ is the temporal rate of change of occupancy of Ca^{2+} binding sites. K_U and K_R are the rate constants for Ca^{2+} uptake and release, respectively. The initial condition for O_c is set, so that $\frac{dO_c}{dt}$ is zero in darkness.

Diffusion between microvillus and somata might also act as a fast free Ca^{2+} shunting. The rate of Ca^{2+} flux from microvillus to somata could be calculated as $\frac{DA}{L}[Ca^{2+}]_i$, whereas $D = 220 \mu m^2/s$ is diffusivity; $L = 60 nm$ is length of somata-microvillus membrane neck; $A = 962 nm^2$ is crossing area of somata-microvillus membrane neck. The rate of Ca^{2+} flux could come out as $10^6 ions/s$ if $[Ca^{2+}]_i$ were to rise above $10 mM$ (coinciding with previous published estimations $8 \mu M$ - $22 mM$ [14]). Although there are physiological measurements showing that $[Ca^{2+}]_i$ can peak at $200 \mu M$, decaying with characteristic time scale of $100 ms$ [12], these experiments were done with blowfly in bright condition. Furthermore, $[Ca^{2+}]_i$ may be underestimated by the assumption that all microvilli were stimulated. The amount of diffused Ca^{2+} is comparable to the rate of Ca^{2+} influx at the peak response, so Ca^{2+} diffusion to somata could not be omitted. Ca^{2+} inside microvillus could diffuse $\sim 1 \mu m$ in $1 ms$. Here, the diffusion time is estimated as $2\sqrt{D\Delta t}$: D is the diffusivity, and Δt is the diffusion time interval, which is much less than light response interval. Thus, $[Ca^{2+}]_i$ is assumed to be uniform in the volume of microvillus during light response. Ca^{2+} diffusion is included in the Ca^{2+} dynamics as a regression term, therefore we have our Ca^{2+} dynamics formulated as in Eq. [12]:

$$\frac{d[Ca^{2+}]_i}{dt} = \frac{I_{Ca,net}}{2\nu_{Ca}F} - n[B]_i \frac{dO_c}{dt} - K_{Ca}[Ca^{2+}]_i. \quad (12)$$

where $[Ca^{2+}]_i$ dynamic is a balance between net Ca^{2+} influx (first term), Ca^{2+} uptake by Ca^{2+} buffer, calmodulin (second term), and Ca^{2+} diffusion (third term). In the second term, n is the number of Ca^{2+} binding sites for calmodulin, here $n = 4$. $[B]_i$ denotes concentration of calmodulin inside the microvillus.

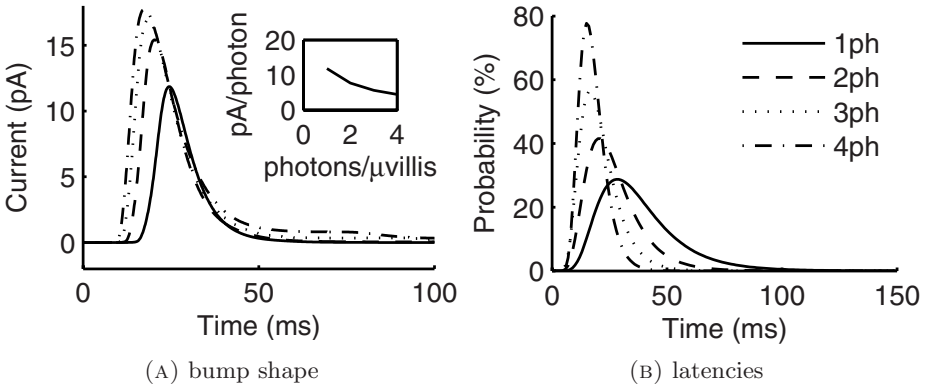


Fig. 5. Signal transduction capability at different light level. (A) Basic bump shape when a single microvillus is absorbing 1, 2, 3, 4 photons, the inset shows peak of bump as a function of number of photons absorbed. (B) Average latencies when a single microvillus is absorbing 1, 2, 3, 4 photons. (A) and (B) share the same legend.

Figs. 5A and Fig. 5B, are to show the different signal transduction capability of a single microvillus when it is absorbing different numbers of photons at

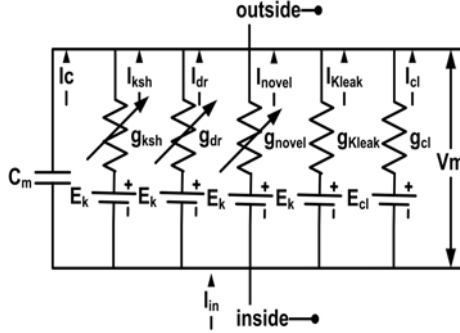


Fig. 6. Electrical circuit of the photoreceptor cell body. Abbreviations: ksh, Shaker channel; dr, delayed rectifier channel; novel, Novel K^+ channel; Kleak, potassium leak conductance; cl, chloride leak conductance.

the same time. It shows that the more photons are absorbed, the less current is produced per photon (the stronger negative feedbacks at brighter light condition; this enables the photoreceptor to effectively use its limited voltage range) and the briefer the latency (the faster are the reactions).

3.5 Model for Latency Dispersion

To overcome the limitations of the deterministic model, which can not describe the variations of signal transduction in different microvilli, we simulated the phototransduction model (Eq. 1 to Eq. 6) stochastically using Gillespie' algorithm. This gives a latency dispersion (time variations in generation of single bumps in different microvilli). For simplicity, we ignore the randomness of the amplitude of different transduction events and assume the randomness only reside in the latencies. The algorithm is from [15]. After simulating phototransduction cascade stochastically for many times, a statistical latency, which is defined as the time for the opening of the first *TRP* channel, can be obtained. For this, we count the number of emerged bumps in each time bin (histogram of latencies), and use a log-normal function to approximate the statistical latency. Latency distribution is obtained by normalizing the log-normal fit.

3.6 Hodgkin-Huxley Model for Photoreceptor Cell Body

Drosophila photoreceptor express three dominant voltage-sensitive K^+ channels in their photo-insensitive membrane (cell body): shaker and two classes of delayed rectifier that differ in their voltage dependency and rate of inactivation [1]. The resulting activation of voltage-sensitive K^+ channels will extrude K^+ out, and thus oppose light-induced depolarization, driving the membrane toward the dark resting potential.

The model for the photoreceptor cell body was based on Hodgkin-Huxley-formalism (for derivation and validation of the model, refer to [10], supplementary material). The model incorporated Shaker and slow delayed rectifier K^+

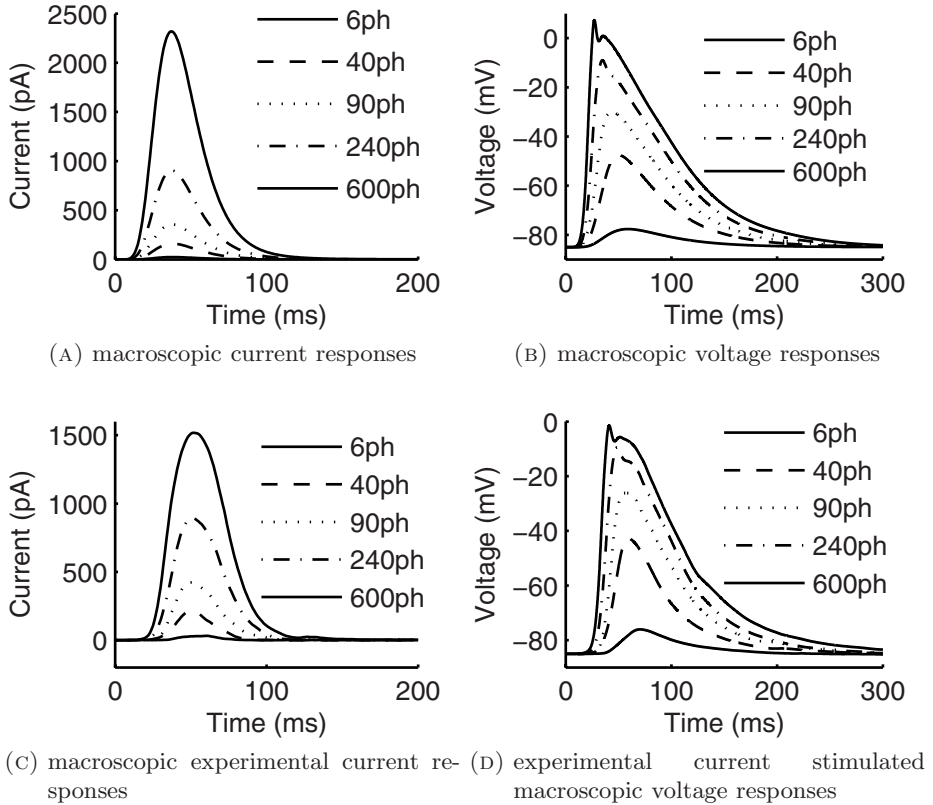
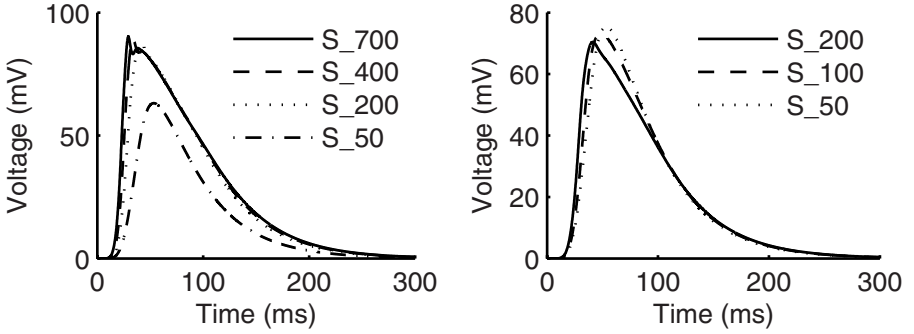


Fig. 7. Simulation results for the model at different light level. (A) Simulated macroscopic current response at light impulse stimuli of 6, 40, 90, 240, 600 photons. (B) Macroscopic voltage responses by the cell body at different level of light impulse stimuli. (C) Experimental macroscopic current responses (patch-clamp data from whole cell recordings) at the same light level shown in Fig. 7A. (D) Voltage response predictions by the model when stimulated by experimental current data.

conductances, in addition to K^+ and Cl^- leak conductances. The voltage-dependent parameters (including time constants and steady-state functions for activation and inactivation of K^+ conductances) were obtained from published data of dark adapted cells [10,11]. Although the properties of delayed rectifier (shab) K^+ channels are regulated by PIP_2 [6], this modulation is much slower than the impulse response of our model. Other photoreceptor membrane properties - *i.e.* the maximum values of the active conductances, resting potential, leak conductances, and membrane capacitance - were estimated from *in vivo* recordings. Though never been measured physiologically, the leak conductances were included to have the right resting potential. It is possible that the leaks could mimic mean inputs from synaptic feedbacks that currently remain uncharacterized. The voltage-dependent properties of the ion channels, the reversal potentials for each



(A) logarithmic scaled voltage responses (B) square root scaled voltage responses

Fig. 8. Scaled voltage responses for comparison. (A) Voltage responses scaled by an logarithmic gain. S_{700} depicts the voltage response under 700 photons stimuli. S_{400} , S_{200} , S_{50} are the 400, 200, 50 photons stimulated voltage responses that are scaled by $\ln(700)/\ln(400)$, $\ln(700)/\ln(200)$, $\ln(700)/\ln(50)$ respectively. (B) Voltage responses scaled by squared root gain under relatively dim light condition (below 200 photons). S_{200} shows the voltage response under 200 photons stimuli. S_{100} , S_{50} are the 100, 50 photons stimulated voltage responses that are scaled by $\sqrt{200}/\sqrt{100}$, $\sqrt{200}/\sqrt{50}$ respectively.

ion species, and the membrane area were kept fixed within the model. Fig. 6 shows the equivalent electrical circuit for the model, where membrane is modeled as capacitor, the equilibrium potential of different species of ion channels as voltage sources, and different kinds of voltage-gated ion channels as adjustable conductances. Leak channels were modeled as non-adjustable conductances.

The simulated current responses (Fig. 7A) and experimental current (Fig. 7C) responses are very similar in shape. However, the activation and inactivation of the simulated responses are somewhat faster than the experimental ones. This discrepancy might result from the left-shift when approximating the statistical latency with log-normal function, leading to a faster estimate. Nonetheless, the peak of simulated macroscopic current is quite linear with light input (number of photons), about $3 - 4 \text{ pA/photon}$, which is in consistent with published data [3]. Whilst the experimental macroscopic current response to 600 photons stimulation appear nonlinear, this compression might be induced by inefficient voltage-clamp control for large currents. The voltage range is almost the same as in Fig. 7B and Fig. 7D, indicating that the cell body model contains the essential nonlinear parts of the cell body. The faster inactivation phase of the estimated voltage response (Fig. 7B) suggests that a log-normal shaped light-induced current might lack a slower boosting component during the inactivation of light response.

Under our simulation, the macroscopic current is quite linear with light intensities, whereas it is the cell body membrane that is highly nonlinear, contributing the most to the compression of voltage responses under relatively bright light

condition. In Fig. 8A, we compared the voltage responses at different light intensities by scaling them with a logarithmic gain. It could be seen that, above 200 photons stimulation, gain scaled voltage responses are quite similar in amplitude. This means that in relatively bright light condition, in logarithmic scale, voltage responses are linear to light intensities. This logarithmic compression under relatively bright light condition help the cell to use efficiently the relatively small voltage range for coding large different light intensities. From our simulation, this compression could be caused mostly by the properties of the voltage gated K^+ conductances. The logarithmic gain control coding is not obtained under relatively dim light condition (under 200 photons/ms), but can be substituted by a square root relationship (Fig. 8B), indicating that cell body membrane could help to shift the gain control mechanism under different light conditions to help using voltage range effectively.

4 Conclusion

We constructed a mathematical model of *Drosophila* R1-R6 photoreceptor to mimic the relationship between voltage outputs and light impulse inputs. The parameters introduced in the model were fixed, if known from electrophysiological experiments, to make physiological sense. Different parts of the models were validated by comparing simulation results with experimental data. The *LIC* part of the model was validated by comparing the simulation results with *in vitro* patch-clamp data [2] and the cell body model was validated by *in vivo* current injection experiments [10]. Even in this relatively basic form, our model can predict well the waveforms of macroscopic light induced current responses. In the future research, naturalistic light input sequences will be introduced to access the proposed dynamics. The fact that we need to enlarge potassium leak conductance in the current clamp mode to keep voltage responses to light in the right range, indicates there are uncharacterized conductances that facilitate adaptation to varying light levels. Nonetheless, from a practical and systemic point of view, this model can serve as a foundation to a preprocessing module for higher order models of the *Drosophila* visual system that we intend to build due course.

Acknowledgments. We thank A. Pumir for discussion and sharing with Gillespie algorithm, we thank Y. Rudy group and T. Pasi group for discussion of Na^+/Ca^{2+} exchanger model. This work was supported by Biotechnology and Biological Sciences Research Council (BBF0120711 and BBD0019001 to MJ). DC and SAB gratefully acknowledge that this work was supported by the Engineering and Physical Sciences Research Council and the European Research Council. ZS thank The University of Sheffield for Ph.D funding.

References

1. Hardie, R.C.: Voltage-sensitive potassium channels in *Drosophila* photoreceptors. *Journal of Neuroscience* 11, 3079–3095 (1991)
2. Hardie, R.C.: Whole-cell recordings of the light induced current in dissociated *Drosophila* photoreceptors: Evidence for feedback by calcium permeating the light-sensitive channels. *Proceedings: Biological Sciences* 245, 203–210 (1991)

3. Hardie, R.C., Postma, M.: Phototransduction in microvillar photoreceptors of *Drosophila* and other invertebrates. *The Senses: A Comprehensive Reference* 1, 77–130 (2008)
4. Hochstrate, P., Hamdorf, K.: Microvillar components of light adaptation in blowflies. *Journal of General Physiology* 95, 891–910 (1990)
5. Juusola, M., Hardie, R.C.: Light adaptation in *drosophila* photoreceptors: I. response dynamics and signaling efficiency at 25 °c. *Journal of General Physiology* 117, 3–25 (2001)
6. Krause, Y., Krause, S., Huang, J., Liu, C.-H., Hardie, R.C., Weckström, M.: Light-dependent modulation of shab channels via phosphoinositide depletion in *Drosophila* photoreceptors. *Neuron*. 59, 596–607 (2008)
7. Liu, C.H., Satoh, A.K., Postma, M., Huang, J., Ready, D.F., Hardie, R.C.: ca^{2+} dependent metarhodopsin inactivation mediated by calmodulin and ninac myosin iii. *Neuron*. 59, 778–789 (2008)
8. Luo, C.H., Rudy, Y.: A dynamic model of the cardiac ventricular action potential: I. simulations of ionic currents and concentration changes. *Circulation Research* 74, 1071–1096 (1994)
9. Mullins, L.J.: A mechanism for na^+/ca^{2+} transport. *Journal of General Physiology* 70, 681–695 (1977)
10. J.E. Niven, M. Vähäsöyrinki, M. Kauranen, R.C. Hardie, M. Juusola, and M. Weckström. The contribution of shaker k^+ channels to the information capacity of *Drosophila* photoreceptors. *Nature* 6923, 630–634 (2003)
11. Oberwinkler, J.C.: Calcium influx, diffusion and extrusion in fly photoreceptor cells. PhD thesis, University of Groningen (2000)
12. Oberwinkler, J.C., Stavenga, D.G.: Light dependence of calcium and membrane potential measured in blowfly photoreceptors in vivo. *Journal of General Physiology* 112, 113–124 (1998)
13. Peretz, A., Abitbol, I., Sobko, A., Wu, C.F., Attali, B.: A ca^{2+} /calmodulin-dependent protein kinase modulates *Drosophila* photoreceptor k^+ currents: A role in shaping the photoreceptor potential. *Journal of Neuroscience* 18, 9153–9162 (1998)
14. Postma, M., Oberwinkler, J.C., Stavenga, D.G.: Does ca^{2+} reach millimolar concentrations after single photon absorption in *Drosophila* photoreceptor microvilli? *Biophys. J.* 77, 1811–1823 (1999)
15. Pumir, A., Graves, J., Ranganathan, R., Shraiman, B.I.: Systems analysis of the single photon response in invertebrate photoreceptors. *Proc. Natl. Acad. Sci. U.S.A* 105, 10354–10359 (2008)
16. Rasmusson, R.L., Clark, J.W., Giles, W.R., Robinson, K., Clark, R.B., Shibata, E.F., Campbell, D.L.: A mathematical model of electrophysiological activity in a bullfrog atrial cell. *American Journal of Physiology - Heart and Circulatory Physiology* 259, 370–389 (1990)
17. van Hateren, J.H., Snippe, H.P.: Phototransduction in primate cones and blowfly photoreceptors: Different mechanisms, different algorithms, similar response. *J. Comp. Physiol. A Neuroethol. Sens. Neural. Behav. Physiol.* 192, 187–197 (2006)
18. Wolff, T., Ready, D.F.: *The Development of Drosophila melanogaster*. Cold Spring Harbor Laboratory Press, Plainview (1993)
19. Wong, F., Knight, B.W., Dodge, F.A.: Dispersion of latencies in photoreceptors of limulus and the adapting-bump model. *Journal of General Physiology* 76, 517–537 (1980)

Comparing a Cognitive and a Neural Model for Relative Trust Dynamics

S. Waqar Jaffry and Jan Treur

VU University Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
{swjaffry, treur}@few.vu.nl
<http://www.few.vu.nl/~{swjaffry, treur}>

Abstract. Trust dynamics can be modelled in relation to experiences. Both cognitive and neural models for trust dynamics in relation to experiences are available, but were not yet related or compared in more detail. This paper presents a comparison between a cognitive and a neural model. As each of the models has its own specific set of parameters, with values that depend on the type of person modelled, such a comparison is nontrivial. In this paper a comparison approach is presented that is based on mutual mirroring of the models in each other. More specifically, for given parameter values set for one model, by automated parameter estimation processes the most optimal values for the parameter values of the other model are determined to show the same behaviour. Roughly spoken the results are that the models can mirror each other up to an accuracy of around 90%.

Keywords: trust dynamics, cognitive, neural, comparison, parameter tuning.

1 Introduction

A variety of computational models has been proposed for the dynamics of human trust in relation to experiences; see e.g., [1-4]. Usually such models consider experiences and trust as cognitive concepts, and depend on values for a set of parameters for specific (cognitive) characteristics of a person, such as trust flexibility vs. rigidity. Recently also neural models for trust dynamics have been introduced. An example of such a neural model, in which in addition a role for emotional responses is incorporated, is described in [5]. Also the latter model includes a specific set of parameters for (neurological) characteristics of the person modelled. As the set of parameters of this neural model has no clear connection to the parameters in cognitive models such as in [4], and the behaviour of such models strongly depends on the values for such parameters, a direct comparison is impossible.

Therefore in this paper, a more indirect way to compare the models is used, by mutual mirroring them in each other. This mirroring approach uses any set of values that is assigned to the parameters for one of the models to obtain a number of simulation traces. These simulation traces are approximated by the second model, based on automated parameter estimation. The error for this approximation is considered as a

comparison measure. In this paper this mirroring approach is applied to the two models for the dynamics of relative trust described in [4] and [5]. It is applied in two directions, and also back and forth sequentially by using the estimated parameter values for the second model to estimate new parameter values for the first.

In the paper, first in Section 2 the cognitive model is briefly summarised, and in Section 3 the neural model. In Section 4 the mirroring approach is discussed and the automated parameter estimation method. Section 5 reports the outcome of some of the experiments performed. Finally, Section 6 is a discussion.

2 A Cognitive Model for the Dynamics of Relative Trust

The cognitive model taken from [4] is composed from two models: one for the positive trust, accumulating positive experiences, and one for negative trust, accumulating negative experiences. First the positive trust is addressed. The human's relative positive trust on an option i at time point t is based on a combination of two parts: the *autonomous* part, and the *context-dependent* part. For the latter part an important indicator is $\tau_i^+(t)$: the ratio of the human's trust of option i to the average human's trust on all options at time point t . Similarly the human's relative negative trust of option i at time point t ($\tau_i^-(t)$) is the ratio between human's negative trust of the option i and the average human's negative trust of the options at time point t . These are calculated as follows:

$$\tau_i^+(t) = \frac{T_i^+(t)}{\sum_{j=1}^n T_j^+(t)/n} \quad \tau_i^-(t) = \frac{T_i^-(t)}{\sum_{j=1}^n T_j^-(t)/n}$$

Here the denominators express the average positive and negative trust over all options at time point t . The context-dependent part is designed in such a way that when the positive trust is above the average, then upon each positive experience it gets an extra increase, and when it is below average it gets a decrease. This principle is a variant of a 'winner takes it all' principle, which for example is sometimes modelled by mutually inhibiting neurons. This principle has been modelled by basing the change of trust upon a positive experience on $\tau_i^+(t) - 1$, which is positive when the positive trust is above average and negative when it is below average. To normalise, this is multiplied by a factor $T_i^+(t) * (1 - T_i^+(t))$. For the autonomous part the change upon a positive experience is modelled by $1 - T_i^+(t)$. As η indicates in how far the human is autonomous or context-dependent in trust attribution, a weighted sum is taken with weights η and $1-\eta$ respectively. Therefore, using the parameters defined in above change in T_i^+ is modelled by the following differential equation:

$$\frac{dT_i^+(t)}{dt} = \beta * [(\eta * (1 - T_i^+(t)) + (1 - \eta) * (\tau_i^+(t) - 1) * T_i^+(t) * (1 - T_i^+(t))) * E_i(t) * (1 + E_i(t))] / 2 - \gamma * T_i^+(t) * (1 + E_i(t)) * (1 - E_i(t))$$

Similarly, for negative trust:

$$\frac{dT_i^-(t)}{dt} = \beta * [\eta * (1 - T_i^-(t)) + (1 - \eta) * (\tau_i^-(t) - 1) * T_i^-(t) * (1 - T_i^-(t))] * E_i(t) * (1 - E_i(t)) / 2 - \gamma * T_i^-(t) * (1 + E_i(t)) * (1 - E_i(t))$$

The trust $T_i(t)$ of option i at time point t is a number between $[-1, 1]$ where -1 and 1 represent minimum and maximum values of the trust respectively. It is the difference of the human's positive and negative trust of option i at time point t : $T_i(t) = T_i^+(t) - T_i^-(t)$. For more details, see [4].

3 A Neural Model for Relative Trust and Emotion

Cognitive states of a person, such as sensory or other representations often induce emotions felt within this person, as described by neurologist Damasio [6] and [7]. Emotion generation via a body loop roughly proceeds according to the following causal chain:

cognitive state \rightarrow preparation for the induced bodily response \rightarrow induced bodily response \rightarrow
sensing the bodily response \rightarrow sensory representation of the bodily response \rightarrow induced feeling

As a variation, an 'as if body loop' uses a direct causal relation preparation for the induced bodily response \rightarrow sensory representation of the induced bodily response as a shortcut in the causal chain. The body loop (or as if body loop) is extended to a recursive body loop (or recursive as if body loop) by assuming that the preparation of the bodily response is also affected by the state of feeling the emotion: feeling \rightarrow preparation for the bodily response as an additional causal relation. Such recursiveness is also assumed by Damasio ([7], pp. 91-92), as he notices that what is felt by sensing is actually a body state which is an internal object, under control of the person. Another neurological theory addressing the interaction between cognitive and affective aspects can be found in Damasio's Somatic Marker Hypothesis; cf. [7-10]. This is a theory on decision making which provides a central role to emotions felt. Within a given context, each represented decision option induces (via an emotional response) a feeling which is used to mark the option. For example, a strongly negative somatic marker linked to a particular option occurs as a strongly negative feeling for that option. Similarly, a positive somatic marker occurs as a positive feeling for that option. Somatic markers may be innate, but may also be adaptive, related to experiences ([8] p. 179). In the model used below, this adaptive aspect is modelled as Hebbian learning; cf. [11-13]. Viewed informally, in the first place it results in a dynamical connection strength obtained as an accumulation of experiences over time (1). Secondly, in decision making this connection plays a crucial role as it determines the emotion felt for this option, which is used as a main decision criterion (2). As discussed in the introduction, these two properties (1) and (2) are considered two main functional, cognitive properties of a trust state. Therefore they give support to the assumption that the strength of this connection can be interpreted as a representation of the trust level in the option considered.

The neural model

An overview of the model for how trust dynamics emerges from the experiences is depicted in Fig. 1. How decisions are made, given these trust states is depicted in Fig. 2. These pictures also show representations from the detailed specifications explained below. However, note that the precise numerical relations between the indicated variables V shown are not expressed in this picture, but explained below.

Activation level for preparation of body state: non-competitive case

The emotional response to the person’s mental state in the form of the preparation for a specific bodily reaction (see label LP4 in Figure 1) is modelled in the non-competitive case as follows. Here the mental state comprises a number of cognitive and affective aspects: options activated, experienced results of options and feelings. This specifies part of the loop between feeling and body state. This dynamic property uses a combination function $g(\sigma, \tau, V_1, V_2, V_3, \omega_1, \omega_2, \omega_3)$ including a threshold function. For example,

$$g(\sigma, \tau, V_1, V_2, V_3, \omega_1, \omega_2, \omega_3) = th(\sigma, \tau, V_1 + \omega_2 V_2 + \omega_3 V_3)$$

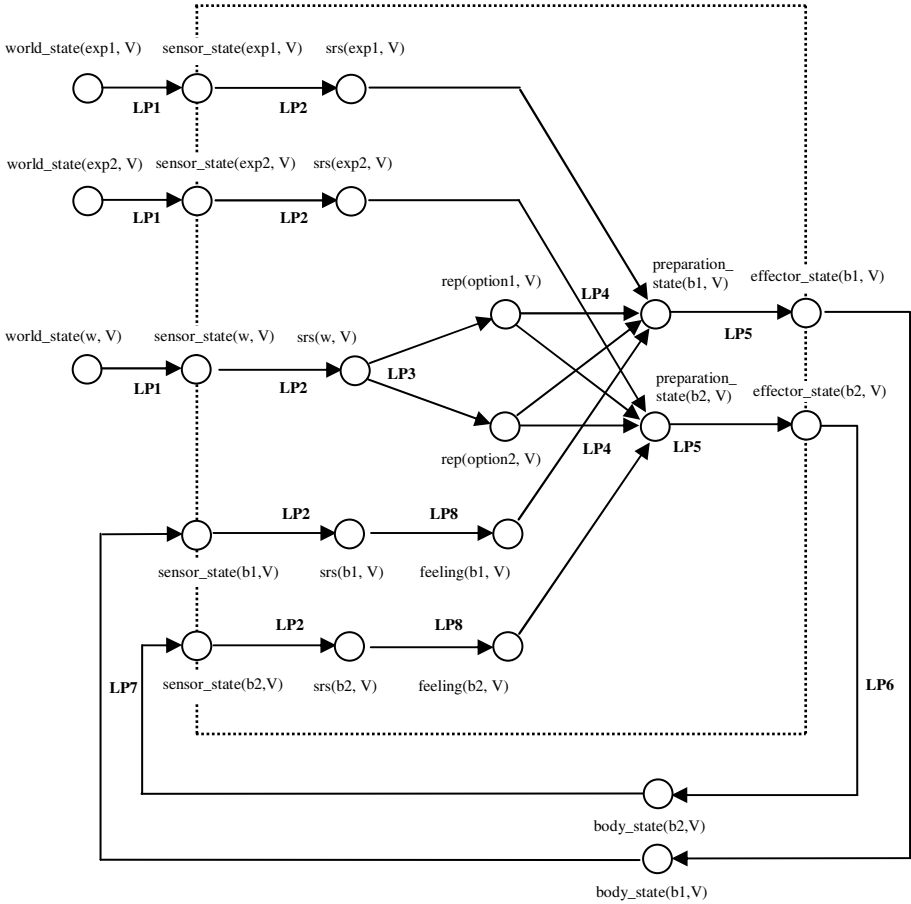


Fig. 1. Overview of the neurological model for dynamics of trust and emotion

with V_1, V_2, V_3 activation levels and $\omega_1, \omega_2, \omega_3$ weights of the connections to the preparation state, and $th(\sigma, \tau, V) = 1/(1 + e^{-\sigma(V-\tau)})$ a threshold function with threshold τ and steepness σ . Then the activation level V_4 of the preparation for an option is modelled by

$$dV_4/dt = \gamma(g(\sigma, \tau, V_1, V_2, V_3, \omega_1, \omega_2, \omega_3) - V_4)$$

Activation level for preparation of body state: competitive case

For the competitive case also the inhibiting cross connections from one represented option to the body state induced by another represented option are used. In this case a function involving these cross connections can be defined, for example for two considered options

$$h(\sigma, \tau, V_1, V_2, V_3, V_{21}, \omega_1, \omega_2, \omega_3, \omega_{21}) = th(\sigma, \tau, \omega_1 V_1 + \omega_2 V_2 + \omega_3 V_3 - \omega_{21} V_{21})$$

with ω_{21} the weight of the suppressing connection from represented option 2 to the preparation state induced by option 1. Then

$$dV_4/dt = \gamma(h(\sigma, \tau, V_1, V_2, V_3, V_{21}, \omega_1, \omega_2, \omega_3, \omega_{21}) - V_4)$$

with V_4 the activation level of preparation for option 1.

Activation level for preparation of action choice

For the decision process on which option O_i to choose, represented by action A_i , a winner-takes-it-all model is used based on the feeling levels associated to the options; for an overview, see label LP10 in Fig. 2.

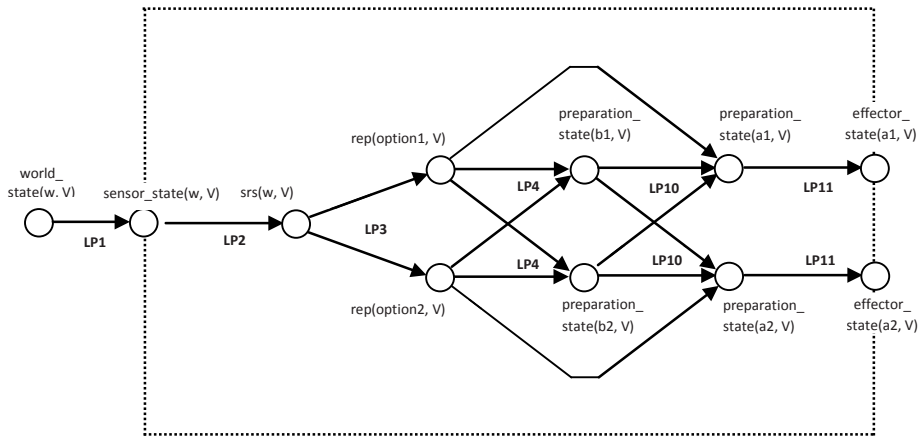


Fig. 2. Overview of the neurological model for trust-based decision making

This has been realised by combining the option representations O_i with their related emotional responses B_i in such a way that for each i the level of the emotional response B_i has a strongly positive effect on preparation of the action A_i related to option O_i itself, but a strongly suppressing effect on the preparations for actions A_j related to the other options O_j for $j \neq i$. As before, this is described by a similar function $h(\sigma, \tau, V_1, \dots, V_m, U_1, \dots, U_m, \omega_{11}, \dots, \omega_{mm})$ as before, with V_i levels for representations of options O_i and U_i levels of preparation states for body state B_i related to options O_i and ω_{ij} the strength of the connection between preparation states for body state B_i and preparation states for action A_j . Based on this, activation level W_i for the preparation of action A_i , is determined by

$$dW_i / dt = \gamma(h(\sigma, \tau, V_1, \dots, V_m, U_1, \dots, U_m, \omega_{11}, \dots, \omega_{mm}) - W_i)$$

The Hebbian adaptation process

From a neurological perspective the strength of a connection from an option to an emotional response may depend on how experiences are felt emotionally, as neurons involved in the option, the preparation for the body state, and in the associated feeling will often be activated simultaneously. Therefore such a connection from option to emotional response may be strengthened based on a general Hebbian learning mechanism [11-13] that states that connections between neurons that are activated simultaneously are strengthened, similar to what has been proposed for the emergence of mirror neurons; e.g., [14] and [15]. This principle is applied to the strength ω_l of the connection from an option to the emotional response expressed by the related body state. The following Hebbian learning rule takes into account a maximal connection strength l , a learning rate η , and an extinction rate ζ .

$$d\omega_l/dt = \eta V_1 V_2 (l - \omega_l) - \zeta \omega_l$$

Here V_1 is the activation level of the option o1 and V_2 the activation level of preparation for body state b1. A similar Hebbian learning rule can be found in ([13] p. 406). By this rule through their affective aspects, the experiences are accumulated in the connection strength from option o1 to preparation of body state b1, and thus serves as a representation of trust in this option o1.

4 The Mirroring Approach to Compare the Models

The mirroring approach used to compare the two parameterised models for trust dynamics works as follows:

- Initially, for one of the models any set of values is assigned to its parameters
- Next, a number of scenarios are simulated based on this first model.
- The resulting simulation traces for the first model are approximated by the second model, based on automated parameter estimation.
- The error for the most optimal values for the parameters of the second model is considered as a comparison measure.

Parameter estimation can be performed according to different methods, for example, exhaustive search, bisection or simulated annealing [16]. As the models considered here have only a small number of parameters exhaustive search is an adequate option. Using this method the entire attribute search space is explored to find the vector of parameter settings with maximum accuracy. This method guarantees the optimal solution, described as follows:

```

for each observed behaviour  $B$ 
  for each vector of parameter value settings  $P$ 
    calculate the accuracy of  $P$ 
  end for
output the vector of parameter settings with maximal accuracy
end for

```

In the above algorithm, calculation of the accuracy of a vector of parameter setting P entails that agent predicts the information source to be requested and observes the

actual human request. It then uses the equation for calculating the accuracy described before. Here if p parameters are to be estimated with precision q (i.e., grain size 10^{-q}), the number of options is n , and m the number of observed outcomes (i.e., time points), then the worst case complexity of the method can be expressed as $O((10)^{pq} nm^2)$, which is exponential in number of parameters and precision. In particular, when $p=3$ (i.e., the parameters β , γ , and η), $q=2$ (i.e., grain size 0.01), $n=3$ and $m=100$, then the complexity will result in 3×10^{10} steps.

5 Comparison Results

A number of experiments were performed using the mutual mirroring approach described in Section 4 to compare the two parameterised models for trust dynamics. Experiments were set up according to two cases:

1. Two competitive options provide experiences *deterministically*, with a constant positive, respectively negative experience, alternating periodically in a period of 50 time steps each (see Fig. 3).
2. Two options provide experiences with a certain *probability* of positivity, again in an alternating period of 50 time steps each.

The first case of experiments was designed to compare the behaviour of the models for different parameters under the same deterministic experiences while the second case is used to compare the behaviour of the models for the (more realistic) case of probabilistic experience sequences. The general configurations of the experiment that are kept constant for all experiments are shown in Table 1.

Table 1. General Experimental Configuration

Parameter	Neural Model	Cognitive Model
Number of competitive options	2	2
Time step (difference equations)	0.1	0.1
Number of time steps	500	500
Initial trust values of option 1 and option 2	0.5, 0.5	0, 0
Strength of connection from option to emotional response (ω_l)	0.5	not applicable
Strength of connection between preparation state of body and preparation state of action (ω_{ij})	0.5	not applicable
Strength of connection between feeling and preparation of body state	0.25	not applicable
Value of the world state	1	not applicable
Grain size in parameter estimation	0.05	0.01

Three experiments were performed for each case: after some parameter values assigned to the cognitive model, its behaviour was approximated by the neural model, using the mirroring approach based on the automatic parameter estimation technique described in Section 4. The best approximating realization of the neural model was

used again to approximate the cognitive model using the same mirroring approach. This second approximation was performed to minimize uni-directionality of the mirroring approach that might bias the results largely if performed from only one model to another and not the other way around.

An instance of a parameterized model can uniquely be represented by a tuple containing the values of its parameters. Here the cognitive and neural models described in Section 2 and 3 are represented by tuples (γ, β, η) and $(\sigma, \tau, \gamma, \eta, \zeta)$ respectively. For the sake of simplicity, a few parameters of the neural model, namely ω_1 , ω_2 and ω_3 , were considered fixed with value 0.5, and were not included in model representation tuple. Furthermore, the initial trust values of both models are assumed neutral (0.0 and 0.5 for cognitive and neural model resp.), see Table 1.

Case 1

In this case the behaviour of the models was compared using the experiences that were provided deterministically with positive respectively negative, alternating periodically in a period of 50 time steps each (see Fig. 3).

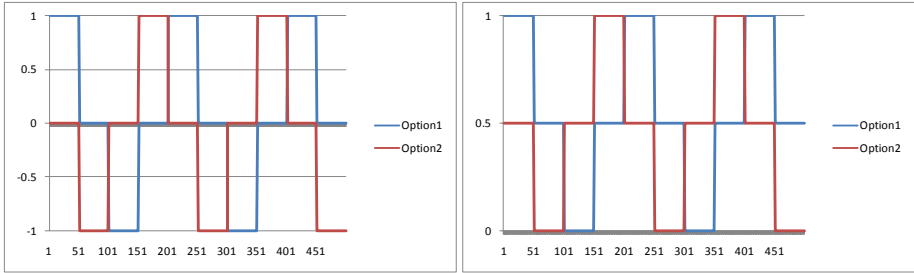


Fig. 3. a) Experience sequence for cognitive model, b) Experience sequence for neural model

Here three different experiments were performed, where the parameters of cognitive model are assigned with some initial values and then its behavior is approximated by the neural model. The best approximation of the neural model against the initially set cognitive model was reused to find the best matching cognitive model. Results of the approximated models and errors are shown in Table 2 while the graphs of the trust dynamics are presented in Fig. 4. Note that for the sake of ease of comparison and calculation of standard error the trust values of cognitive model are projected from the interval $[-1, 1]$ to $[0, 1]$ (see Fig. 4). In Table 2, the comparison error ε is the average of the root mean squared error of trust of all options, as defined by the following formula,

$$\varepsilon = \frac{1}{n} * \sum_{i=1}^n \sqrt{\sum_{j=1}^m (T(j)_{1i} - T(j)_{2i})^2}$$

In the above formulation, n is the number of options, m is the number of time steps while $T(j)_{1i}$ and $T(j)_{2i}$ represent trust value of option i at time point j for each model, respectively.

In Table 2 for experiment 1 initially the cognitive model was set with parameters (0.99, 0.75, 0.75) which was then approximated by the neural model. The best approximation of the neural model was found to be (0.55, 10, 0.15, 0.90, 0.50) with an approximate average of root mean squared error of all options ϵ value 0.074050.

Table 2. Results of Case 1

Exp.	Initial Model	Approximating Model using the mirroring approach	Comparison Error (ϵ)
1	Cog. Mod. (0.99, 0.75, 0.75)	Neu. Mod. (0.55, 10, 0.15, 0.90, 0.50)	0.074050
	Neu. Mod. (0.55, 10, 0.15, 0.90, 0.50)	Cog. Mod. (0.96, 0.20, 0.53)	0.034140
2	Cog. Mod. (0.88, 0.99, 0.33)	Neu. Mod. (0.35, 10, 0.60, 0.95, 0.60)	0.071900
	Neu. Mod. (0.35, 10, 0.60, 0.95, 0.60)	Cog. Mod. (0.87, 0.36, 0.53)	0.059928
3	Cog. Mod. (0.75, 0.75, 0.75)	Neu. Mod. (0.30, 10, 0.95, 0.90, 0.60)	0.138985
	Neu. Mod. (0.55, 10, 0.15, 0.90, 0.50)	Cog. Mod. (0.83, 0.37, 0.55)	0.075991

Then this setting of neural model was used to approximate cognitive model producing best approximate with parameter values (0.96, 0.20, 0.53) producing ϵ 0.034140. Similarly the results of other two experiments can be read in Table 2.

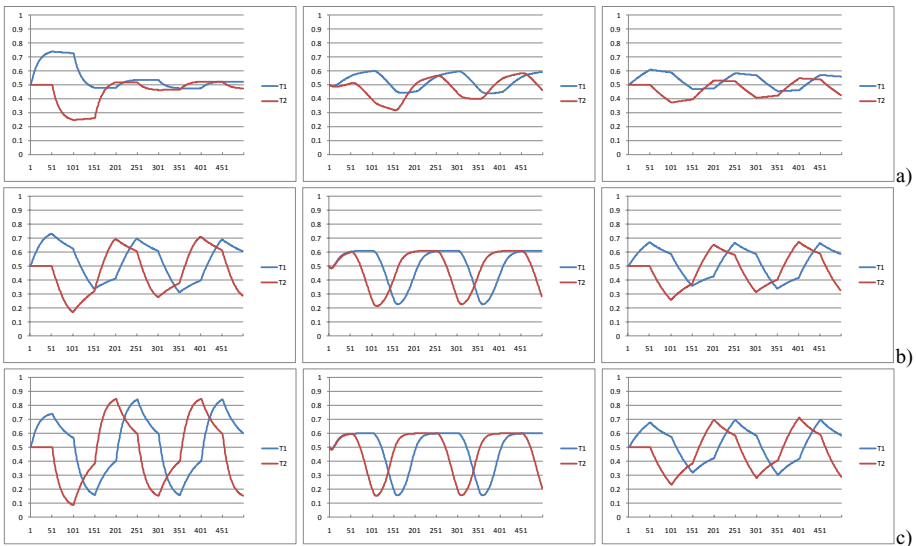


Fig. 4. Dynamics of the Trust in Case 1 a) Experiment 1, b) Experiment 2, c) Experiment 3

Fig. 4 represents the dynamics of the trust in the two options over time for the deterministic case. The horizontal axis represent time step while vertical axis represent the value of trust. The graphs for each experiment are represented as set of three figures, where the first figure shows the dynamics of the trust of both options by the cognitive model with an initial setting as described in the second column of the first row of each experiment of Table 2. The second figure shows the traces of the

dynamics of trust by the neural model as described in the third column of the first row of each experiment of Table 2. Finally the third figure shows the approximation of the cognitive model by the neural model, where the neural model is described in the second column of the second row of each experiment of Table 2 (which is similar to third column of the first row of each experiment), and the approximated cognitive model is presented in the third column of the second row of each experiment. From Table 2 and Fig. 4 it can be observed that the mirroring approach based on automatic parameter estimation when used in bidirectional way gives a better realization of both models in each other, resulting in a smaller comparison error and better curve fit.

Case 2

In the second case the behaviour of the models was compared when experiences are provided with a certain probability of positivity, again in an alternating period of 50 time steps each. Also here three different experiments were performed, where the parameters of the cognitive model were assigned with some initial values and then its behaviour was approximated by the neural model. The best approximation of the neural model against initially set cognitive model was reused to find the best matching cognitive model. In experiment 1, 2 and 3 the option 1 and option 2 give positive experiences with (100, 0), (75, 25) and (50, 50) percent of probability, respectively. Results of approximated models and errors for this case are shown in Table 3 while the graphs of trust dynamics are presented in Fig. 5. Note that for the sake of ease of comparison and calculation of the standard error, again the trust values of the cognitive model are projected from the interval $[-1, 1]$ to $[0, 1]$ (see Fig. 5). In Table 3 for experiment 1 initially the cognitive model was set with parameters (0.99, 0.75, 0.75) which was then approximated by the neural model.

Table 3. Results of Case 2

Exp.	Initial Model	Approximating Model using the mirroring approach	Error (ϵ)
1	Cog. Mod. (0.99, 0.75, 0.75)	Neu. Mod. (0.85, 10, 0.95, 0.20, 0.05)	0.061168
	Neu. Mod. (0.85, 10, 0.95, 0.20, 0.05)	Cog. Mod. (0.97, 0.99, 0.18)	0.045562
2	Cog. Mod. (0.99, 0.75, 0.75)	Neu. Mod. (0.40, 20, 0.90, 0.20, 0.15)	0.044144
	Neu. Mod. (0.40, 20, 0.90, 0.20, 0.15)	Cog. Mod. (0.83, 0.05, 0.99)	0.039939
3	Cog. Mod. (0.99, 0.75, 0.75)	Neu. Mod. (0.10, 20, 0.45, 0.10, 0.10)	0.011799
	Neu. Mod. (0.10, 20, 0.45, 0.10, 0.10)	Cog. Mod. (0.99, 0.50, 0.99)	0.011420

The best approximation of the neural model was found to be (0.85, 10, 0.95, 0.20, 0.05) with an approximate average of root mean squared error of all options ϵ of value 0.061168. Then this setting of neural model was used to approximate cognitive model producing best approximate with parameter values (0.97, 0.99, 0.18) and ϵ 0.034140. Similarly the results of other two experiments could also be read in Table 3.

Fig. 5 represents the dynamics of the trust in the two options over time for the probabilistic case. The horizontal axis represents time while the vertical axis represents the values of trust. Here also the graphs of each experiment are represented as set of three figures, where the first figure shows the dynamics of the trust in both options by the cognitive model with an initial setting as described in the second column of the first row of each experiment of Table 3.

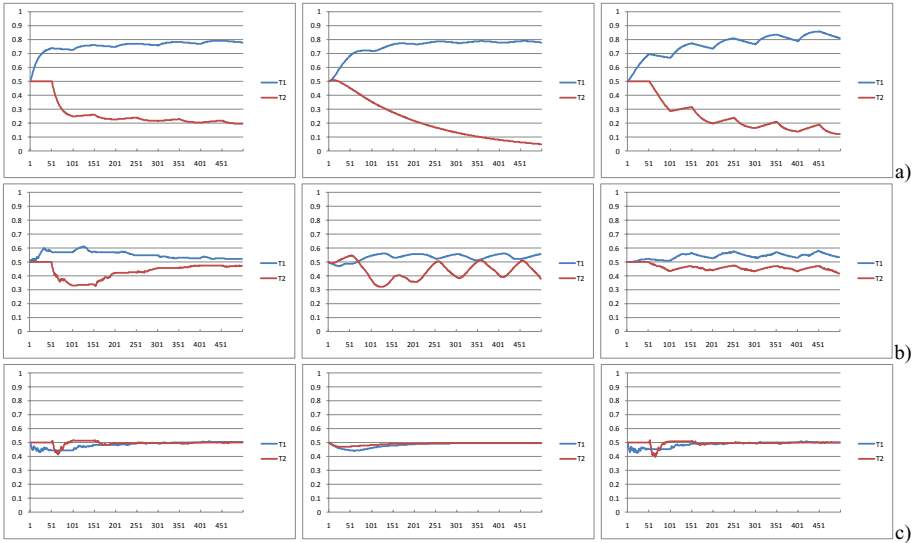


Fig. 5. Dynamics of the Trust in Case 2, a) Experiment 1, b) Experiment 2, c) Experiment 3

The second figure shows the traces of the dynamics of trust by the neural model as described in the third column of the first row of each experiment of Table 3. Finally, the third figure is the approximated cognitive model by the neural model, where the neural model is described in the second column of the second row of each experiment of Table 3 (which is similar to third column of the first row of each experiment), and the approximated model is presented in the third column of the second row of each experiment.

As already noticed in case 1, also here it can be observed that the mirroring approach based on automatic parameter estimation when used in bidirectional way gives a better realization of both models in each other, resulting smaller comparison error and a better curve fit. Furthermore, it can also be noted that as the uncertainty in the options behaviour increases, both models show more similar trust dynamics producing lower error value in comparison.

6 Discussion

In this paper two parameterised computational models for trust dynamics were compared: a cognitive model and a neural model. As the parameter sets for both models are different, the comparison involved mutual estimation of parameter values by which the models were mirrored into each other in the following manner. Initially, for one of the models any set of values was assigned to the parameters of the model, after which a number of scenarios were simulated based on this first model. Next, the resulting simulation traces for this first model were approximated by the second model, based on automated parameter estimation. The error for the most optimal values for the parameters of the second model was considered as a comparison measure. It turned out that

approximations could be obtained with error margins of about 10%. Furthermore the results for the (more realistic) case of probabilistic experience sequences have shown much better approximation than for the deterministic case. This can be considered a positive result, as the two models have been designed in an independent manner, using totally different techniques. In particular, it shows that the cognitive model, which was designed first, without taking into account neurological knowledge, can still be grounded in a neurological context, which is a nontrivial result.

References

1. Jonker, C.M., Treur, J.: Formal Analysis of Models for the Dynamics of Trust based on Experiences. In: Garijo, F.J., Boman, M. (eds.) MAAMAW 1999. LNCS (LNAI), vol. 1647, pp. 221–232. Springer, Heidelberg (1999)
2. Jonker, C.M., Treur, J.: A Temporal-Interactivist Perspective on the Dynamics of Mental States. *Cognitive Systems Research Journal* 4, 137–155 (2003)
3. Falcone, R., Castelfranchi, C.: Trust dynamics: How Trust is Influenced by Direct Experiences and by Trust Itself. In: Proc. of AAMAS 2004, pp. 740–747 (2004)
4. Hoogendoorn, M., Jaffry, S.W., Treur, J.: Modeling Dynamics of Relative Trust of Competitive Information Agents. In: Klusch, M., Pěchouček, M., Polleres, A. (eds.) CIA 2008. LNCS (LNAI), vol. 5180, pp. 55–70. Springer, Heidelberg (2008)
5. Hoogendoorn, M., Jaffry, S.W., Treur, J.: Modelling Trust Dynamics from a Neurological Perspective. In: Proceedings of the Second International Conference on Cognitive Neurodynamics, ICCN 2009. Springer, Heidelberg (to appear, 2009)
6. Damasio, A.: *The Feeling of What Happens. Body and Emotion in the Making of Consciousness*. Harcourt Brace, New York (1999)
7. Damasio, A.: *Looking for Spinoza*. Vintage books, London (2004)
8. Damasio, A.: *Descartes' Error: Emotion, Reason and the Human Brain*. Papermac, London (1994)
9. Damasio, A.: The Somatic Marker Hypothesis and the Possible Functions of the Prefrontal Cortex. *Philosophical Transactions of the Royal Society: Biological Sciences* 351, 1413–1420 (1996)
10. Bechara, A., Damasio, A.: The Somatic Marker Hypothesis: a neural theory of economic decision. *Games and Economic Behavior* 52, 336–372 (2004)
11. Hebb, D.: *The Organisation of Behavior*. Wiley, New York (1949)
12. Bi, G.Q., Poo, M.M.: Synaptic Modifications by Correlated Activity: Hebb's Postulate Revisited. *Ann. Rev. Neurosci.* 24, 139–166 (2001)
13. Gerstner, W., Kistler, W.M.: Mathematical formulations of Hebbian learning. *Biol. Cybern.* 87, 404–415 (2002)
14. Keysers, C., Perrett, D.I.: Demystifying social cognition: a Hebbian perspective. *Trends in Cognitive Sciences* 8, 501–507 (2004)
15. Keysers, C., Gazzola, V.: Unifying Social Cognition. In: Pineda, J.A. (ed.) *Mirror Neuron Systems: the Role of Mirroring Processes in Social Cognition*, pp. 3–28. Humana Press/Springer Science (2009)
16. Hoogendoorn, M., Jaffry, S.W., Treur, J.: An Adaptive Agent Model Estimating Human Trust in Information Sources. In: Proceedings of the 9th IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2009. IEEE Computer Society Press, Los Alamitos (to appear, 2009)

A Next Generation Modeling Environment PLATO: Platform for Collaborative Brain System Modeling

Shiro Usui^{1,2}, Keiichiro Inagaki², Takayuki Kannon¹,
Yoshimi Kamiyama³, Shunji Satoh⁴, Nilton L. Kamiji¹,
Yutaka Hirata⁶, Akito Ishihara⁷, and Hayaru Shouno⁵

¹ Laboratory for Neuroinformatics, RIKEN Brain Science Institute

² Computational Science Research Program, RIKEN

³ School of Information Science and Technology, Aichi Prefectural University

⁴ Graduate School of Information Systems, The University of
Electro-Communications

⁵ Department of Information and Communication Engineering, The University of
Electro-Communications

⁶ Department of Computer Science, Chubu University

⁷ School of Information Science and Technology, Chukyo University
usuishiro@riken.jp

<http://www.ni.brain.riken.jp/>

Abstract. To understand the details of brain function, a large scale system model that reflects anatomical and neurophysiological characteristics needs to be implemented. Though numerous computational models of different brain areas have been proposed, these integration for the development of a large scale model have not yet been accomplished because these models were described by different programming languages, and mostly because they used different data formats. This paper introduces a platform for a collaborative brain system modeling (PLATO) where one can construct computational models using several programming languages and connect them at the I/O level with a common data format. As an example, a whole visual system model including eye movement, eye optics, retinal network and visual cortex is being developed. Preliminary results demonstrate that the integrated model successfully simulates the signal processing flow at the different stages of visual system.

Keywords: Neuroinformatics, Model integration, Large scale modeling, Visual system, Common data format.

1 Introduction

The brain presides essential roles of human life and fulfills precise and flexible processing generated by its complicated network. To elucidate the signal processing carried out by the network, numerous neuroscience researches have

been conducted in a large variety of the field such as anatomy, neurophysiology, molecular biology, immunochemistry, and computational science. The multimodal approaches in the neuroscience research have revealed a great deal of function in the brain network. However, the resources obtained in experiments and modeling studies have not been shared among neuroscientists, but mainly as published articles. To further elucidate the brain function systematically, it should be replicated as a precise large scale system model; for instance the numerous resources in the field mentioned above and related computational models should be integrated.

A primary role for neuroinformatics is to archive numerous digital resources in neuroscience — for instance, experimental data and scripts of computational models — and to share them among worldwide neuroscientists using open access databases [1,2,3,16,17]. Under the International Neuroinformatics Coordinating Facility (INCF), we have also established the neuroinformatics Japan-node and neuroscience platforms [2] where physiological data, analysis tools, and computational models are registered and shared among neuroscientists. In order to support this trend, a simulation server platform is being developed [4] as one of the Japan-node platforms. On the simulation server platform, researchers can simulate and confirm results of models stored in the platforms. As described above, the neuroscience databases and the use of them are being developed; however, a framework for the integration of models registered in the neuroscience databases for the developing the whole brain system has not yet been designed.

Here we propose a next generation modeling environment named PLATO (Platform for a coLLaborative brAin sysTem mOdeling) [6]. In the PLATO, computational models can be constructed by using several resources (e.g. experimental data, articles, and models), several programming languages, and connecting them at the I/O level with the Network Common Data Form (netCDF) [7] to build a large scale system model. In developing the model, the resources are collected from among neuroscience databases including the neuroinformatics Japan-node and neuroscience platforms with a data management tool (Concierge) [5]. In the present work, we introduce more detail of the system configuration of PLATO and a novel function library which assists to program model I/Os. As a test case for the PLATO, a large scale visual system model including eye movement, eye optics and retinal network, is being developed. Preliminary results are introduced below.

2 Framework of the PLATO

2.1 System Configuration of PLATO

The PLATO consists of a data management tool (Concierge), modeling tools and simulation servers (Fig. 1A). The Concierge is a personal database tool for managing digital research resources [5]: articles, physiological data, analysis programs and computational models, including those registered data on servers such as the neuroinformatics platforms in Japan-node [2]. Once the resources for developing a model (e.g. articles and experimental data) are collected with the

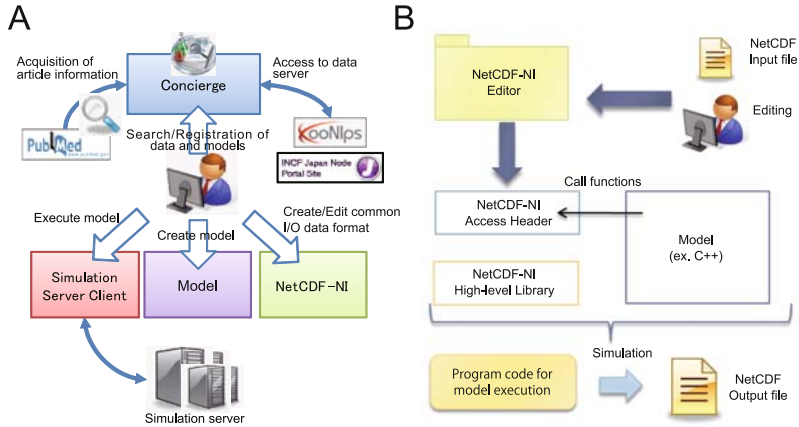


Fig. 1. System configuration of the PLATO and an example of modeling on the PLATO environment. A: The PLATO consists of data management tool (Concierge), modeling tools (simulators and NetCDF-NI) and simulation server. B: Procedure of modeling and simulation on the PLATO divided into creation and editing a common I/O data format by NetCDF-NI, coding a model with a NetCDF-NI access header library and its high-level library and simulation.

Concierge, the user can develop a model using several programming languages (e.g. C/C++, MATLAB, Python and Java) and simulators on the PLATO, and run them on the PLATO simulation server.

2.2 Network Common Data Form and NetCDF-NI

In order to integrate models developed by different programming language to construct a large scale model, model I/Os should be described in the same manner: that is, simulation step size, data dimension and data unit. The PLATO recommends and produces a common I/O data format known as netCDF. The netCDF format can include data and metadata such as simulation step size, data dimension, data units, and equations of a model; therefore the netCDF file itself produces all the necessary information about the input or output of a model. Moreover, models can be pluggable by using the netCDF format. The netCDF format is independent of the operating system, and libraries for fast parallel data access are also available. These schemes will be essential to integrate models developed by different programming languages in different computer architectures and run it on a parallel-processor computer system.

To facilitate the use of netCDF, we are developing the NetCDF-NI: a GUI based software that can create a netCDF access header library including variables and its metadata (e.g. variables, its unit and data formats) that are required for a model and its I/O configuration. It also contains several functions to access the netCDF data files.

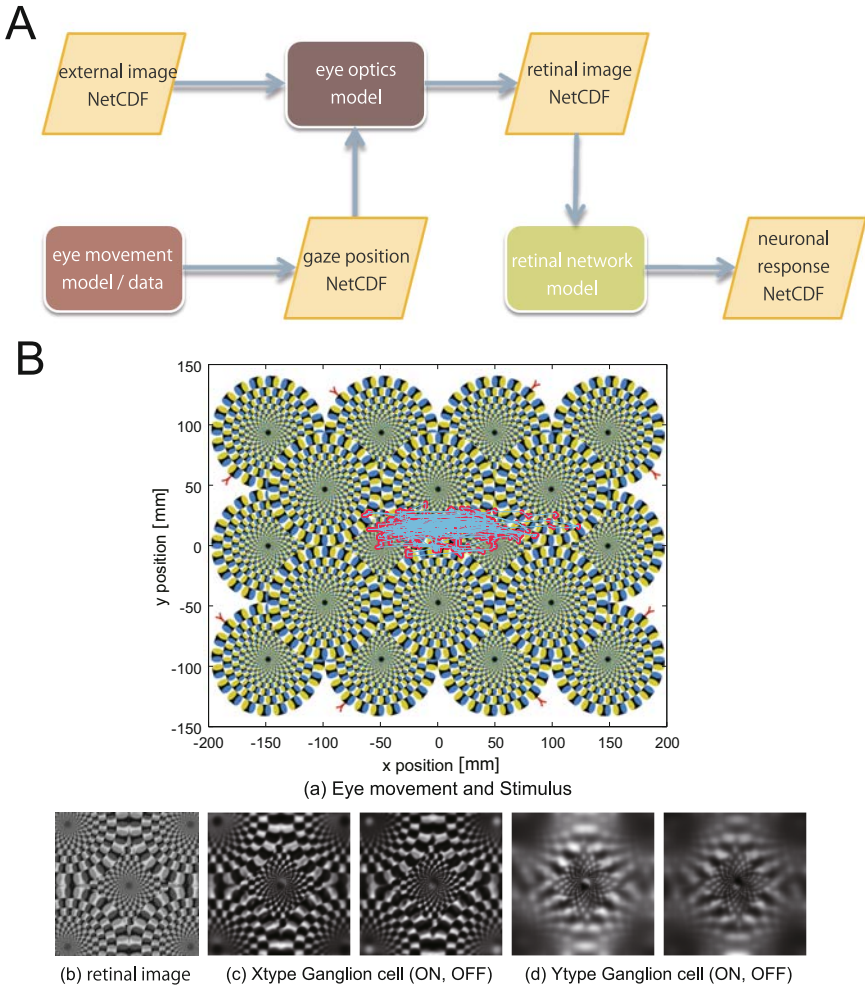


Fig. 2. An example of model development and integration on the PLATO. (A) the schematic diagram of model integration; (B) simulation results: (a) rotation snake stimulus and eye movement (red dots: fixating point, cyan line: scan path), a snapshot of (b) retinal image, (c) X-type ganglion cells response and (d) Y-type ganglion cells response.

2.3 Procedure for Modeling on the PLATO

Fig. 2B illustrates a procedure for developing a model on the PLATO. It consists of (1) creating and editing of a netCDF file by using the NetCDF-NI, (2) coding a model with a netCDF access header and its high-level library, and (3) simulation. In creating and editing the netCDF file, users can generate a netCDF access header library using the NetCDF-NI by defining variables and its metadata, which are used in a model and its I/O. The models can be developed by C/C++,

MATLAB, Python and Java on the PLATO environment. The variables and I/O configurations (e.g. data form, unit and simulation step size) are loaded from a netCDF file via the netCDF access header library. The developed models also tested on the PLATO simulation server.

3 Evaluation of the PLATO

To demonstrate our image of the PLATO, we preliminarily constructed a visual system model including eye movement, eye optics, and retinal network. Fig. 2A [6] illustrates a schematic diagram of the visual system model constructed on the PLATO. In the model, each of the system, an eye movement model or data, an eye optics model and a retinal network model, were connected with the netCDF format. Due to the characteristics of the netCDF format either the eye movement model output or the experimental eye movement data could be plugged in. In other words, the PLATO supports the use of physiological data into the computational model. The eye optics model was improved basing upon Artal's model [8] taking account of recent evidences for the eye ball: architectures [9] and optic characteristics such as accommodation [10], pupil diameter [11] and spectral transmittance [12,13] to calculate a retinal image. The retinal image was further processed in a retinal network model. We utilized the virtual retina [14] as a preliminary model. The virtual retina replicated the functions of the retinal network and computed the activities of X-type and Y-type ganglion cell output.

Fig. 2B summarizes the simulation results produced by the visual system model. A rotating snake stimulus [15] and eye movements (red dots: fixating point, cyan line: scan path) are summarized in Fig. 2B (a), and a snapshot of model outputs in Fig. 2B (b-d). These results demonstrated that the model successfully reproduced the signal processing at the different stages of the visual system: an external image which was acquired by eye movements is successfully converted to a retinal image by the eye optics model (b); then the retinal network model could generate on/off X type ganglion cells (c) and Y type ganglion cells (d) output.

4 Summary and Conclusion

In the present work we proposed a novel modeling environment PLATO and demonstrated that the large scale visual system model was successfully integrated, as confirmed by the visualization of the signal processing flow at the different stages of the visual system. These results indicated that the netCDF data format could be a good bridge of the model I/O described by different format in the integration of models.

For further improvements of the PLATO, we are currently developing a function library of the netCDF format to automatically adjust the simulation step size between the model I/Os. We hope that the function library will allow users to freely integrate their models on the PLATO.

The PLATO tightly collaborates with the neuroinformatics platforms available in the Japan-node, because numerous models and physiological data are continuously being registered. Likewise, it can be made possible to utilize the resources registered on the other neuroscience databases by implementing plug-ins to the Concierge. That is, the PLATO can provide multidisciplinary modeling environment by this collaboration. Finally, we hope that the PLATO will help researchers to develop models and to integrate them for constructing a large scale brain model in near future.

Acknowledgments. This research was supported by “The Next-Generation Integrated Simulation of Living Matters”, part of the Development and Use of the Next-Generation Supercomputer Project of the Ministry of Education, Culture, Sports, Science and Technology.

References

1. Bjaalie, J.G., Grillner, S., Usui, S.: Neuroinformatics: Databases, tools, and computational modeling for studying the nervous system. *Neural Networks* 21, 1045–1046 (2008)
2. Usui, S., Furuichi, T., Miyakawa, H., Ikeno, H., Nagao, S., Iijima, T., Kamiyama, Y., Isa, T., Suzuki, R., Ishikane, H.: Japanese neuroinformatics node and platforms. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) *ICONIP 2007, Part II. LNCS*, vol. 4985, pp. 884–894. Springer, Heidelberg (2008)
3. Usui, S., Okumura, Y.: Basic scheme of neuroinformatics platform: XooNips. In: Zurada, J.M., Yen, G.G., Wang, J. (eds.) *Computational Intelligence: Research Frontiers. LNCS*, vol. 5050, pp. 102–116. Springer, Heidelberg (2008)
4. Usui, S., Yamazaki, T., Ikeno, H., Okumura, Y., Satoh, S., Kamiyama, Y., Hirata, Y., Inagaki, K., Kannon, T., Kamiji, N.L., Ishihara, A.: Simulation platform: a test environment of computational models via web. In: *INCF Congress of Neuroinformatics, Pilsen, Czech Republic*, pp. 943–972 (submitted 2009)
5. Sakai, H., Aoyama, T., Yamaji, K., Usui, S.: Concierge: personal database software for managing digital research resources. *Frontiers in neuroinformatics* 1–5, 1–6 (2007)
6. Inagaki, K., Kannon, T., Kamiyama, Y., Satoh, S., Kamiji, N.L., Hirata, Y., Ishihara, A., Shouno, H., Usui, S.: Platform for collaborative brain system modeling (PLATO): toward large scale modeling for visual system. In: *INCF Congress of Neuroinformatics, Pilsen, Czech Republic*, pp. 943–128 (2009)
7. NetCDF user’s guide. <http://www.unidata.ucar.edu/software/netcdf/>
8. Artal, P.: Calculations of two-dimensional foveal retinal images in real eyes. *J. Opt. Soc. Am. A* 7(8), 1374–1381 (1990)
9. Stiles, W.S., Crawford, B.H.: The luminous efficiency of rays entering the eye pupil at different points. *Proc. R. Soc. Lond. B* 112, 428–450 (1933)
10. L’opez-Gil, N., Iglesias, I., Artal, P.: Retinal image quality in the human eye as a function of the accommodation. *Vis. Res.* 38, 2897–2907 (1998)
11. Schwiegerling, J.: Scaling Zernike expansion coefficients to different pupil sizes. *J. Opt. Soc. Am. A* 19, 1937–1945 (2002)
12. Xu, J., Pokorny, J., Smith, V.C.: Optical density of the human lens. *J. Opt. Soc. Am. A* 14, 953–960 (1997)

13. Stockman, A., Sharpe, L.T.: Spectral sensitivities of the middle- and long-wavelength sensitive cones derived from measurements in observers of known genotype. *Vis. Res.* 40, 1711–1737 (2000)
14. Wohrer, A., Kornprobst, P.: Virtual Retina: a biological retina model and simulator, with contrast gain control. *J. Comp. Neurosci.* 26(2), 219–249 (2009)
15. Kitaoka, A., Ashida, H.: Phenomenal characteristics of the peripheral drift illusion. *Vision* 15, 261–262 (2003)
16. Kamper, L., Bozkurt, A., Rybacki, K., Geisser, A., Gerken, I., Stephen, K.E., Kötter, R.: An introduction to CoCoMac-Online. The online-interface of the primate connectivity database CoCoMac. In: *Neuroscience database. A practical guide*, pp. 155–169. Kluwer Academic Publishers, Boston (2003)
17. Hines, M.L., Morse, T., Migliore, M., Carnevale, N.T., Shepherd, G.M.: ModelDB: A Database to Support Computational Neuroscience. *J. Comp. Neurosci.* 7, 7–11 (2003)

Modeling Geomagnetospheric Disturbances with Sequential Bayesian Recurrent Neural Networks

Lahcen Ouarbya and Derrick T. Mirikitani

Department of Computing, Goldsmiths College, University of London, New Cross,
London SE14 6NW

Abstract. Sequential Bayesian trained recurrent neural networks (*RNNs*) have not yet been considered for modeling the dynamics of magnetospheric plasma. We provide a discussion of the state-space modeling framework and an overview of sequential Bayesian estimation. Three nonlinear filters are then proposed for online *RNN* parameter estimation, which include the extended Kalman filter, the unscented Kalman filter, and the ensemble Kalman filter. The exogenous inputs to the *RNNs* consist of three parameters, b_z , b^2 , and b_y^2 , where b , b_z , and b_y represent the magnitude, the southward and azimuthal components of the interplanetary magnetic field (*IMF*) respectively. The three models are compared to a model used in operational forecasts on a severe double storm that has so far been difficult to forecast. It is shown that some of the proposed models significantly outperform the current state of the art.

Keywords: Geomagnetic Storms, Recurrent Neural Networks, Filtering.

1 Introduction

It has been well established that changes in the Sun's magnetic field influences the structure of the magnetic field surrounding the earth (Geo-magnetosphere) [13, 8]. The solar wind¹ expands the reach of the Sun's magnetic field to form what is known as the Interplanetary Magnetic Field (*IMF*). The *IMF* can cause energetic particles to be injected into the Earth's magnetic field, resulting in Geo-magnetospheric disturbances. Disruption of the Geo-magnetosphere takes place when a transfer of energy from the solar wind opposes the Earth's magnetic field. A magnetospheric storm occurs if this transfer of energy persists for several hours [8]. Geomagnetic storms can have many negative effects on technical systems in space and on Earth, such as a change in a spacecraft orientation, terrestrial power generation and transmission².

Forecasts of the earth's magnetic field can give vital information about the intensity of future magnetospheric disturbances. At mid-latitudes, magnetic storms are measured in terms of the horizontal component of the Earth's magnetic field [8]. This horizontal component is averaged to form an index known as D_{st} . Studies have shown

¹ Solar winds are a stream of charged particles, mostly electrons and protons, that are ejected from the upper atmosphere of the sun.

² A nuclear generator belonging to the OKG utility company in Sweden was heated from geomagnetically induced current caused by the magnetic storm of March 13-14, 1989.

a correlation between the intensity of magnetic storms and the value of the D_{st} index [9][6], where the more negative the D_{st} index the greater the intensity of the magnetic storm. The physical interaction (transfer of mass, energy and momentum) between the IMF and the Geo-magnetosphere takes place at the magnetopause boundary. The interaction itself is not fully understood and thus previous researchers have built non-parametric predictive models usually based on recurrent neural networks ($RNNs$) to forecast the D_{st} index [12][13]. Recurrent neural networks were found to uncover some of the relationships between the IMF and D_{st} well enough for real time forecasts [14].

Previous work in modeling the relationship between IMF and D_{st} with $RNNs$ have relied heavily on first-order gradient based methods for parameter estimation of the model which has resulted in long training times [12][19], uncertain convergence, and possibly vanishing gradients. This has been a bottleneck in the area (and may stifle future progress in neural based forecasting of geomagnetic phenomena), as well performing models are difficult to obtain, and new events can not readily be incorporated into the model for improved forecasts. In this paper we investigate solutions to this problem through the use of the sequential Bayesian framework of which nonlinear Kalman filters are utilized for RNN training [20]. The advantage of our approach is a framework based on second-order [22] online estimation of model parameters, resulting in fast convergence and accurate forecasts. The main results of the paper are as follows: 1) an efficient framework to reliably obtain RNN parameters for D_{st} forecasts, 2) the ability to sequentially incorporate new measurements into the model, 3) improved forecast accuracy over previously demonstrated results.

2 Recurrent Neural Networks

In this study, the recurrent architecture known as the Elman network (RNN) [4] is chosen as previous studies have found a successful results with $RNNs$. Feed-forward networks are not considered in this study due to poor performance in modeling the recovery phase dynamics [7]. This is mostly likely due to the limitation of the feed-forward architecture, i.e. limited temporal memory, bounded by the dimension of the input window.

The Elman RNN consists of a feed-forward multi-layer perceptron architecture, augmented with a context layer which connects every hidden neuron output to every hidden neuron input. The context layer allows for a memory of past states of the network. The network weights for the hidden layer of size H can be represented as a matrix defined as

$$\mathbf{W}_h = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_H]^T \quad (1)$$

where $\mathbf{w}_i = [w_{i,0}, w_{i,1}, \dots, w_{i,j}]^T$ $i = 1, 2, \dots, H$, $j = I + H$, and I is the size of the input layer. The hidden state $\mathbf{s}(t)$ is connected to the network output $y(t) \in \mathbb{R}^1$ via the output weight vector

$$\mathbf{w}_{out} = [w_0, w_1, \dots, w_H]^T \quad (2)$$

The operation of the Elman network is described by the following discrete time equations:

$$\begin{aligned} \mathbf{s}(t) &= \mathbf{g}\left(\mathbf{W}_h[b, \mathbf{x}(t)^T, \mathbf{c}(t)^T]\right) \\ \mathbf{y}(t) &= \mathbf{f}(\mathbf{w}_{out}^T[b, \mathbf{s}(t)^T])^T \end{aligned} \quad (3)$$

where $\mathbf{c}(t) = \mathbf{s}(t-1) \in \mathbb{R}^H$ is the context vector, $\mathbf{x}(t)$ is the exogenous *IMF* inputs, and b is the bias. The functions $\mathbf{g}(\cdot)$ and $\mathbf{f}(\cdot)$ are typically logistic sigmoidal nonlinearities $\sigma(a) = 1/(1 + \exp(-a))$ which map the input a from \mathbb{R} into a bounded interval $\Omega = (0, 1)$ of length $|\Omega| = 1$ where $\Omega \subset \mathbb{R}$. All weights [1](#) and [2](#) can be arranged into one vector as follows

$$\mathbf{w}(t) = [\mathbf{w}_1^T, \mathbf{w}_2^T, \dots, \mathbf{w}_H^T, \mathbf{w}_{out}^T]^T \quad (4)$$

2.1 State Space Modeling with RNNs

Weight estimation of the *RNN* can be formulated in a sequential Bayesian filtering framework: given a hidden state represented by *RNN* weights and a noise contaminated measurement, the task is to re-estimate the weights so as to factor in the newly arrived information. The weights in the recurrent neural network, $\mathbf{w}(t) \in \mathbb{R}^p$, are considered as the discretized state of the system. The *RNN* weights $\mathbf{w}(t)$ are treated as a random vector whose time evolution is specified by the following nonlinear discrete time state space model

$$\begin{aligned} \mathbf{w}(t) &= \mathbf{w}(t-1) + \boldsymbol{\omega}(t-1) \quad \textit{process equation} \\ d(t) &= h(t, \mathbf{w}(t), \mathbf{x}(t)) + \nu_t \quad \textit{measurement equation} \end{aligned} \quad (5)$$

where $\boldsymbol{\omega}(t) \in \mathbb{R}^p$ represents a stochastic perturbation assumed to be an i.i.d. (independent and identically distributed) Gaussian process with zero mean and covariance \mathbf{Q} , i.e. $\boldsymbol{\omega}(t) \sim \mathcal{N}(0, \mathbf{Q})$, and $t \in \mathbb{N}$ is the time index. This error $\boldsymbol{\omega}(t)$ represents the discrepancy between the *RNN* and the underlying state transition function. The noise $\nu_t \in \mathbb{R}^1$ is assumed to be independent, zero-mean, uncorrelated, Gaussian with variance R : $\nu(t) \sim \mathcal{N}(0, R)$, and $d(t)$ are the targets from the provided data set $D = \{\mathbf{x}(t), d(t)\}_{t=1}^T$. The measurement equation represents the overall behavior of the *RNN*, and the associated error, represented by $\nu(t)$, models the noise in the observations.

In the sequential filtering framework, it is assumed that past information $P(\mathbf{w}(t-1)|\mathbf{d}(t-1))$ is available and can be used to find two quantities of interest: $P(\mathbf{w}(t)|\mathbf{d}(t-1))$, the forecast (prior) distribution and $P(\mathbf{w}(t)|\mathbf{d}(t))$, which is the analysis (posterior) distribution. The forecast distribution is specified via the integral

$$P(\mathbf{w}(t)|\mathbf{d}(t-1)) = \int P(\mathbf{w}(t)|\mathbf{w}(t-1))P(\mathbf{w}(t-1)|\mathbf{d}(t-1))d\mathbf{w}(t-1) \quad (6)$$

The posterior distribution is filtered using the Bayes rule, which combines the prior information $P(\mathbf{w}(t)|\mathbf{d}(t-1))$ with the most recently observed information $P(d(t)|\mathbf{w}(t))$ to compute the analysis distribution

$$P(\mathbf{w}(t)|\mathbf{d}(t)) \approx P(d(t)|\mathbf{w}(t))P(\mathbf{w}(t)|\mathbf{d}(t-1)) \quad (7)$$

A sequential estimation of the two distributions is achieved through iteration of this cycle at each time step.

3 EKF Training of the RNN

For RNN training, the state space equations are highly nonlinear, and must be linearized before the Kalman filter equations can be computed at each time step [10]. This linearization before applying the Kalman filter is known as the Extended Kalman Filter (EKF). The real time recurrent learning algorithm was used for the linearization of the RNN through computation of the Jacobian matrix $\mathbf{H}(t) = \partial h(\cdot)/\partial \mathbf{w}(t)$ consisting of partial derivatives of the output $\mathbf{y}(t)$ with respect to the weights of the network. The Jacobian $\mathbf{H}(t)$ is evaluated at each time step. EKF filtering for RNN weight estimation leads to faster convergence [21] than gradient based algorithms, and also may resolve issues with vanishing gradients [10]. For neural networks, The EKF solution to the parameter estimation problem is given by the following recursion

$$\begin{aligned} \mathbf{k}_g(t+1) &= \mathbf{P}(t)\mathbf{H}(t+1)[R + \mathbf{H}(t+1)^T\mathbf{P}(t)\mathbf{H}(t+1)]^{-1} \\ \mathbf{w}(t+1) &= \mathbf{w}(t) + \mathbf{k}_g(t+1)(\mathbf{d}(t) - h(t, \mathbf{w}(t), \mathbf{x}(t))) \\ \mathbf{P}(t+1) &= \mathbf{P}(t+1) - \mathbf{k}_g(t+1)\mathbf{H}(t+1)^T\mathbf{P}(t). \end{aligned} \tag{8}$$

Since the EKF is a suboptimal estimator based on linearization of a nonlinear mapping, $\mathbf{w}(t)$ is only an approximation of the expectation, $\mathbf{P}(t)$ is an approximation of the state covariance, and the matrix $\mathbf{k}_g(t)$ is the Kalman gain. It is well known that the EKF may experience instabilities as a result of this approximation, especially in situations of high nonlinearity.

4 UKF Training of the RNN

The shortcomings of EKF [10] have lead many researchers to develop a number of closely related Gaussian approximate filters based on novel deterministic sampling methods used to propagate Gaussian random variables. [11] have introduced a more robust alternative, the Unscented Kalman filter (UKF). Unlike the EKF, UKF propagates mean and covariance information through nonlinear transformation using the unscented transform (UT). Let \mathbf{x} be an L -by-1 random variable with $\hat{\mathbf{x}}$ and $\mathbf{P}^{\mathbf{xx}}$ its mean and covariance respectively. Let \mathbf{y} be the transform of \mathbf{x} through a nonlinear function, $\mathbf{y} = f(\mathbf{x})$. Let χ be a matrix of $2L + 1$ sigma vectors χ_i used to approximate the random variable \mathbf{x} .

The weights for the calculation of the posteriori mean and covariance as follows

$$\begin{aligned} \mathbf{W}_0^{(m)} &= \frac{\lambda}{L+\lambda} \\ \mathbf{W}_0^{(c)} &= \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \\ \mathbf{W}_i^{(m)} &= \mathbf{W}_i^{(c)} = \frac{1}{2(L+\lambda)} \quad i = 1 \dots 2L \end{aligned} \tag{9}$$

where $\mathbf{W}_i^{(m)}$ and $\mathbf{W}_i^{(c)}$ represent the mean and covariance weights respectively. The parameter $\lambda = \alpha^2(L + k) - L$ represents a scaling parameter where L is the length of the state vector and the value of α is often between 0.001 and 1. The parameter k represents a secondary scaling parameter, usually set to $3 - L$. The parameter β represents information about prior knowledge of the distribution of \mathbf{x} .

After each iteration of the the *UKF* the sigma points are calculated as follows

$$\Gamma(t) = (L + \lambda)(\mathbf{P}(t) + \mathbf{Q}(t)) \quad (10)$$

$$\phi^i(t) = [(\hat{\mathbf{w}}(t))_{i=0}, (\hat{\mathbf{w}}(t) + \sqrt{\Gamma(t)})_{1 \leq i \leq L}, (\hat{\mathbf{w}}(t) - \sqrt{\Gamma(t)})_{L < i \leq 2L}] \quad (11)$$

$$\mathbf{D}^i(t) = \mathbf{h}(t, \phi^i(t), \mathbf{x}(t)) \quad \mathbf{y}(t) = \mathbf{h}(t, \hat{\mathbf{w}}(t), \mathbf{x}(t)) \quad (12)$$

where $\mathbf{P}(t)$ and $\mathbf{Q}(t)$ represent L-by-L approximate error covariance and process noise covariance matrices, respectively.

The sigma points $\{\chi_i\}_{i=0}^{2L}$ are propagated through the nonlinear function $\mathbf{y}_i = f(\chi_i)$, where $i = 0, \dots, 2L$. We then use weighted average of the transformed sigma points to approximate the mean of \mathbf{y} , $\hat{\mathbf{y}}$, as follows

$$\hat{\mathbf{y}} = \sum_{i=0}^{2L} \mathbf{W}_i^{(m)} \mathbf{y}_i \quad (13)$$

The weighted average of the difference between each transformed sigma point and the overall average is used to calculate the predicted error covariance as follows

$$\mathbf{P}^{\mathbf{y}\mathbf{y}} = \sum_{i=0}^{2L} \mathbf{W}_i^{(c)} (\mathbf{y}_i - \hat{\mathbf{y}})(\mathbf{y}_i - \hat{\mathbf{y}})^T \quad (14)$$

The *RNN-UKF* weight vector is then updated online as follows

- The filtered measurement estimate error covariance matrix and the cross covariance between the state and measurement

$$\begin{aligned} \mathbf{P}^{\mathbf{y}\mathbf{y}}(t) &= \sum_{i=0}^{2L} \mathbf{W}_i^{(c)} (\mathbf{D}^i(t) - \mathbf{y}(t))(\mathbf{D}^i(t) - \mathbf{y}(t))^T \\ \mathbf{P}^{\mathbf{w}\mathbf{y}}(t) &= \sum_{i=0}^{2L} \mathbf{W}_i^{(c)} (\phi^i(t) - \hat{\mathbf{w}}(t))(\mathbf{D}^i(t) - \mathbf{y}(t))^T \end{aligned} \quad (15)$$

- The gain matrix, the filtered state estimate and the error covariance are computed as follows

$$\begin{aligned} \mathbf{k}_g(t) &= \mathbf{P}^{\mathbf{w}\mathbf{y}}(t)(\mathbf{P}^{\mathbf{y}\mathbf{y}}(t) + R)^{-1} \\ \hat{\mathbf{w}}(t+1) &= \hat{\mathbf{w}}(t) + \mathbf{k}_g(t)(d(t) - h(t, \hat{\mathbf{w}}(t), \mathbf{x}(t))) \\ \mathbf{P}(t+1) &= \mathbf{P}(t) - \mathbf{k}_g(t)\mathbf{P}^{\mathbf{y}\mathbf{y}}\mathbf{k}_g(t)^T \end{aligned} \quad (16)$$

5 Ensemble Kalman Filter Training of the *RNN*

The *EnKF* is a Markov Chain Monte Carlo approach to estimating the Fokker-Plank equation for the time evolution of the PDF of the *RNN* weights [5]. The *EnKF* samples the PDF by a random ensemble of *RNN* weights, which approximates the ensemble density [16]. The *EnKF* algorithm can be broken down in to two phases, the prediction step and the analysis step. In the prediction step, the state gets propagated forward in time via Equation 5. As stated in Section 2, the state vector is defined as

$\mathbf{w}(t) = [w_1, w_2, \dots, w_n]^T$, where n is the number of parameters of the *RNN*. An ensemble of size m is formed by drawing m of these \mathbf{w}_t vectors from $\mathcal{N}(0, \mathbf{Q})$ to form the ensemble matrix $\boldsymbol{\theta}(t) = [\mathbf{w}^1(t), \mathbf{w}^2(t), \dots, \mathbf{w}^i(t), \dots, \mathbf{w}^m(t)]$, $\boldsymbol{\theta}(t) \in \mathbb{R}^{(n \times m)}$ in which $\boldsymbol{\theta}^{(i)}(t) = \mathbf{w}^i(t)$. The ensemble mean can then be written as $\bar{\boldsymbol{\theta}}(t) = \boldsymbol{\theta}(t) \cdot \mathbf{1}_{(m \times m)}$ where $\mathbf{1}_{(m \times m)}$ is a $(m \times m)$ matrix where all elements in $\mathbf{1}_{(m \times m)}$ are set to the value of $1/m$. Furthermore, $y_t^i = h(\boldsymbol{\theta}^{(i)}(t))$ where $\mathbf{y}(t) = [y^1(t), y^2(t), \dots, y^i(t), \dots, y^m(t)]$. The mean of the forecast ensemble can then be defined as $\bar{\mathbf{y}}(t) = \mathbf{y}(t) \mathbf{1}_{(m \times m)}$ and, the covariance matrixes are estimated as

$$\mathbf{P}_{\mathbf{w}\mathbf{y}}(t) = \frac{1}{m-1}(\boldsymbol{\theta}(t) - \bar{\boldsymbol{\theta}}(t))(\mathbf{y}(t) - \bar{\mathbf{y}}(t))^T \quad (17)$$

$$\mathbf{P}_{\mathbf{y}\mathbf{y}}(t) = \frac{1}{m-1}(\mathbf{y}(t) - \bar{\mathbf{y}}(t))(\mathbf{y}(t) - \bar{\mathbf{y}}(t))^T \quad (18)$$

The mean weight vector is computed as $\bar{\mathbf{w}}(t) = \boldsymbol{\theta}(t) \cdot \mathbf{1}_{(m \times 1)}$ where again $\mathbf{1}_{(m \times 1)}$ is a $(m \times 1)$ matrix where all elements in $\mathbf{1}_{(1 \times m)}$ are set to the value of $1/m$. The weights are then updated by

$$\bar{\mathbf{w}}(t+1) = \bar{\mathbf{w}}(t) + \mathbf{P}_{\mathbf{w}\mathbf{y}}(t)(R + \mathbf{P}_{\mathbf{y}\mathbf{y}}(t))^{-1}(d(t) - h(\bar{\mathbf{w}}(t), x(t))) \quad (19)$$

Finally, to update the ensemble, a random matrix $\mathbf{Z} \in \mathbb{R}^{(m \times n)}$ is created where each element z_{ij} is drawn from $\mathcal{N}(0, R)$. The mean of the sample is computed via $\bar{\mathbf{Z}} = \mathbf{Z} \cdot \mathbf{1}_{(m \times m)}$, and the variance is computed via

$$\mathbf{D} = \sqrt{\frac{m}{m-1}}(\mathbf{Z} - \bar{\mathbf{Z}}) \quad (20)$$

where the ensemble is then updated as

$$\mathbf{G}(t) = (\boldsymbol{\theta}(t) - \bar{\boldsymbol{\theta}}(t)) + \mathbf{P}_{\mathbf{w}\mathbf{y}}(t)(R + \mathbf{P}_{\mathbf{y}\mathbf{y}}(t))^{-1}(\mathbf{D} - (\mathbf{y}(t) - \bar{\mathbf{y}}(t))) \quad (21)$$

$$\boldsymbol{\theta}(t+1) = \mathbf{G}(t) + \bar{\mathbf{w}}(t+1)\mathbf{c}_{(1 \times m)} \quad (22)$$

and $\mathbf{c}_{(1 \times m)}$ is a $(1 \times m)$ matrix of ones. The analysis step adds a random component and perturbs the ensemble that will be used as the starting point for the next iteration of the algorithm. This update step propagates the mean and variance of the ensemble forward, updated in light of the new observation.

6 Experimental Results

We compare the forecast performance of the three proposed models and an operational model [14] on one hour ahead forecasting of the D_{st} index. The data set consisted of 216 hourly measurements taken from 7-Nov-2004 to 15-Nov-2004.

In all simulations, the weights of the networks were initialized with random uniformly distributed weights in the range of $[-1, 1]$. Each of the Kalman trained recurrent networks were initialized with 3 hidden neurons and 3 input neurons for each factor

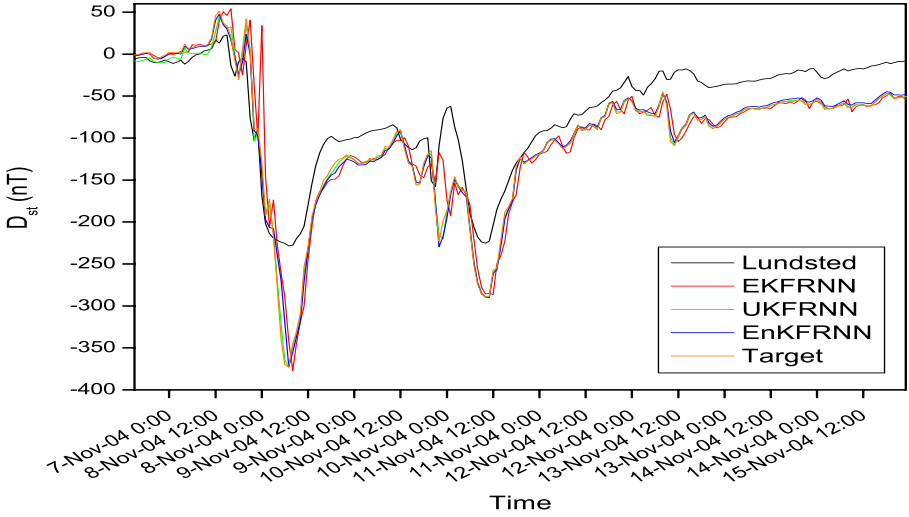


Fig. 1. Prediction of D_{st} from 7-Nov-2004 to 15-Nov-2004

(b_z, b^2, b_y^2) . All RNNs had one output neuron corresponding to the one hour ahead value of the D_{st} signal. In all filters, the initial diagonal elements of the covariance matrix of the $[Q]_{ii}$ and R were set to $1.0e^{-3}$ and $1.0e^{-2}$.

Figure 1 show that the forecasts of the *RNN-EnKF* and the *RNN-UKF* are similar and the top two graphs of Figure 2 show that the errors of the *RNN-EnKF* and *RNN-UKF* are clustered around zero, with few large errors. The *EKF* trained RNN and Lundstedt of-line model [15] resulted in the least accurate predictions, with Lundstedt's model producing the largest errors, as shown in the bottom graph of Figure 2. The relatively poor performance of the *RNN-EKF* is most likely due to filter divergence, as large errors are committed during the sharp downward spike of the D_{st} index. However, the sample based filters captured this sudden change with little error. Lundstedt's model is not dynamically updated and has resulted in poor forecast performance [14]. From these simulations it is clear that online training of RNNs with nonlinear Kalman filters can significantly improve D_{st} forecasts respectively.

7 Future Work

Neural networks have shown encouraging results in modeling D_{st} . The research presented here demonstrate the ability of dynamically trained RNNs to accurately capture the behavior of D_{st} . However, on the main limitations of the work is the static topology of the network (parameters are adapted, but not the topology), which does not allow for fully adaptive adjustment of the neural model to the underlying process. Further work will evaluate more flexible models such the removal of weights or basis functions [17], and generating topologies to suit the data [2][18].

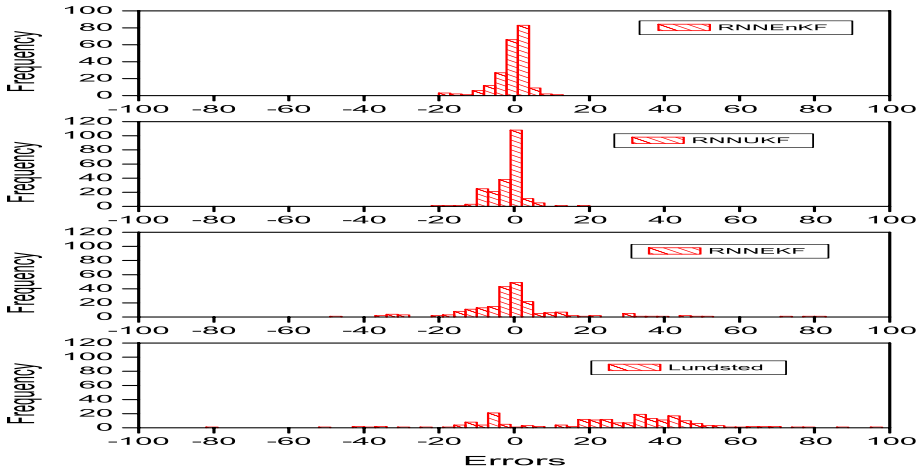


Fig. 2. Distribution of errors

8 Concluding Remarks

This paper introduced a framework for recursive estimation of geomagnetic activity through sequential Bayesian training of RNNs. The proposed filters implement a stable online equivalent of second-order Newtons Method for parameter estimation [21]. Through a comparison between the proposed models and [15], we have observed a significant increase in prediction accuracy of the Bayesian filtered RNNs. The advantage of the online filters is due to second order training of the RNN with precisely adapted learning rates computed in the Kalman gain of the filters. It is known that Kalman based training methods contain second order information of the scaled inverse Hessian of the cost function, which results in increased learning, and robustness to local minima during weight optimization. The geomagnetic forecasting literature has heavily utilized first order gradient based methods. This paper has shown significant improvements with second order Kalman filter trained RNNs.

References

1. Axford, W.I., Hines, C.O.: A unifying theory of high-latitude geophysical phenomena and geomagnetic storms. *Can. J. Phys.* 39, 1433–1464 (1961)
2. de Menezes, L., Nikolaev, N.: Forecasting with Genetically Programmed Polynomial Neural Networks. *International Journal of Forecasting* 22(2), 249–265 (2005)
3. Dungey, J.W.: Interplanetary magnetic field and the auroral zones. *Phys. Rev. Lett.* 26, 47–48 (2000)
4. Elman, J.L.: Finding Structure in Time. *Cognitive Science* 14, 179–211 (1990)
5. Evensen, G.: The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics* 53, 343–367 (2003)

6. Farrugia, C.J., Freeman, M.P., Burlaga, L.F., Lepping, R.P., Takahashi, K.: The earth's magnetosphere under continued forcing - Substorm activity during the passage of an interplanetary magnetic cloud. *J. Geophys. Res.* 98, 7657–7671 (1993)
7. Gleisner, H., Lundstedt, H., Wintoft, P.: Predicting Geomagnetic Storms From Solar-Wind Data Using Time-Delay Neural Networks. *Ann. Geophys.* 14, 679–686 (1996)
8. Gonzales, W.D., Joselyn, J.A., Kamide, Y., Kroehl, H.W., Rostoker, G., Tsurutani, B.T., Vasyliunas, V.M.: What is a geomagnetic storm? *J. Geophys. Res.* 99, 5771–5792 (1994)
9. Gosling, J.T., McComas, D.J., Phillips, J.L., Bame, S.J.: Geomagnetic activity associated with earth passage of interplanetary shock disturbances and coronal mass ejections. *J. Geophys. Res.* 96, 7831–7839 (1991)
10. Haykin, S.: *Kalman Filtering and Neural Networks*. John Wiley & son, New York (2001)
11. Julier, S., Uhlmann, J.: A New Extension of the Kalman Filter to Nonlinear Systems. In: *Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068, pp. 182–193 (1997)
12. Lundstedt, H.: Neural Networks and prediction of solar-terrestrial effects. *Planet. Space Sci.* 40, 457–464 (1992)
13. Lundstedt, H., Wintoft, P.: Prediction of geomagnetic storms from solar wind data with the use of a neural network. *Ann Geophys.* 12, 19–24 (1994)
14. Lundstedt, H., Gleisner, H., Wintoft, P.: Operational forecasts of the geomagnetic Dst index. *Geophys. Res. Lett.* 29, 34–1–34–4 (2002)
15. Lund Space Weather Center, <http://www.lund.irf.se/rwc/dst/models/>
16. Mirikitani, D.T., Nikolaev, N.: Dynamic Modeling with Ensemble Kalman Filter Trained Recurrent Neural Networks. In: *ICMLA 2008*, pp. 843–848 (2008)
17. Nikolaev, N., de Menezes, L.: Sequential Bayesian Kernel Modelling with Non-Gaussian Noise. *Neural Networks* 21(1), 36–47 (2008)
18. Nikolaev, N., Iba, H.: Polynomial Harmonic GMDH Learning Networks for Time Series Modeling. *Neural Networks* 16(10), 1527–1540 (2003)
19. Pallochia, G., Amata, E., Consolini, G., Marcucci, M.F., Bertello, I.: Geomagnetic Dst index forecast based on IMF data only. *Ann Geophys.* 24, 989–999 (2006)
20. Puskorius, G.V., Feldkamp, L.A.: Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks. *IEEE T Neural Networks* 5, 279–297 (1994)
21. Ruck, D.W., Rogers, S.K., Kabrisky, M., Maybeck, P., Oxle, M.E.: Comparative Analysis of Backpropagation and the Extended Kalman Filter for Training Multilayer Perceptrons. *IEEE Trans. Patt. Anal. & Mach. Intell.* 14(6), 686–691 (1992)
22. Schottky, B., Saad, D.: Statistical mechanics of EKF learning in neural networks. *J. Phys. A: Math Gen.* 32, 1605–1621 (1999)

Finding MAPs Using High Order Recurrent Networks

Emad A.M. Andrews and Anthony J. Bonner

Department of Computer Science
University of Toronto
Toronto, ON
emad@cs.toronto.edu

Abstract. Belief revision is the problem of finding the most plausible explanation for an observed set of evidences. This has many applications in various scientific domains like natural language understanding, medical diagnosis and computational biology. Bayesian Networks (BN) is an important probabilistic graphical formalism used widely for belief revision tasks. In BN, belief revision can be achieved by setting the values of all random variables such that their joint probability is maximized. This assignment is called *the maximum a posteriori* (MAP) assignment. Finding MAP is an NP-Hard problem. In this paper, we are proposing finding the MAP assignment in BN using High Order Recurrent Neural Networks through an intermediate representation of Cost-Based Abduction. This method will eliminate the need to explicitly construct the energy function in two steps, objective and constraints, which will decrease the number of free parameters to set.

1 Introduction

Belief revision is the problem of finding the most plausible explanation for an observed set of evidences. Belief revision falls under the broader domain of reasoning under uncertainty where information is not complete or contradictory; thus, probabilistic handling seemed the best candidate for those tasks. However, probabilistic reasoning was described as being “*epistemologically inadequate*” by McCarthy and Hayes in their basic paper in 1969 [1]. They showed that the number of parameters needed to compute the joint probability distribution is exponentially proportional to the size of the given dataset which yields the whole mathematical computations intractable. As a result, researchers avoided using probabilistic reasoning until the notion of independence assumption appeared.

In 1988, Pearl standardized the independence assumption notion and formally presented Bayesian Network (BN) where each variable is conditionally independent of its ancestors given its parents [2]. BN is fully specified by two components: a Directed Acyclic Graph (DAG), whose vertices represent random variables, and a set of parameters that describe the conditional probability distribution of each variable given its parents. These two components together fully specify a unique joint distribution over all random variables in the graph. Let G be a DAG, and let x_1, \dots, x_n denote the set of random variables, vertices of G . The BN encodes the *Markov assumption*: Each

variable is independent of all its non-descendants variables given its parents. Thus, the full joint distribution can be composed of the product form:

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | \pi(X_i)) \quad (1)$$

$\pi(X_i)$ is the set of parents of X_i in G . For a specific assignment, A , over all nodes, (1) can be rewritten as:

$$P(A) = \prod_{i=1}^n P(A(X_i) | A(\pi(X_i))) \quad (2)$$

BN saves a considerable amount of memory and calculations which enables us to calculate joint distributions otherwise impossible to calculate. For example, to specify the full joint distribution for 10 binary random variables we need $2^{10} = 1024$ values to be stored and used during computation. If we used BN with each variable depending on no more than three other variables, we end up having $10 \times 2^3 = 80$ parameters only.

Given a BN with an observed set of evidence nodes \mathcal{E} we are looking for values assignment A for the rest of the network nodes, such that $P(A | \mathcal{E})$ is maximized, using Bayes rule:

$$P(A | \mathcal{E}) = \frac{P(A)P(\mathcal{E} | A)}{P(\mathcal{E})} \quad (3)$$

Because we have observed the values of evidence nodes \mathcal{E} , $P(\mathcal{E})$ is constant, so it ends up maximizing $P(A)$ that represents the joint probability distribution in (1) and (2). This assignment is called *the maximum a posteriori* assignment (MAP). Once this assignment is found, we can do all kinds of probabilistic inference needed.

Finding MAP is shown to be NP-hard [4]. For multiply-connected BN, existing algorithms suffer from exponential complexity, so new heuristics and algorithms are always needed. In this paper, we propose finding MAP using High Order Recurrent Neural Networks (HORN) through an intermediate representation of Cost-Based Abduction (CBA). We first transform BN into CBA system using the algorithm mentioned in [6]. To our knowledge, this is the only algorithm in literature that does such a transformation, so it is crucial to analyze and discuss it step by step. Then, we will fill in the gap between BN and HORN by solving the resultant CBA system using HORN through the method mentioned in [7]. Finally, we will provide the mathematical framework to derive the energy functions equivalent to logical rules with more than 3 hypotheses and present our results.

1.1 Cost-Based Abduction (CBA)

CBA was first introduced by Charniak et al [9]. Formally, a CBA system is a 4-tuple (H, R, c, G) , where H is a set of hypotheses or propositions, c is a function from H to a nonnegative real $c(h)$ called the assumability cost of $h \in H$, R is a set of rules of

the form: $R:(p_{i_1} \wedge p_{i_2} \wedge \dots \wedge p_{i_n}) \rightarrow p_{i_k}$ for all $p_{i_1}, \dots, p_{i_n} \in H$, and $G \in H$ is the goal or the evidence set [6].

Objective: finding the least cost proof (LCP) of the goal. Proof cost is the sum of all costs of the hypotheses needed to be assumed to complete the proof. Any given hypothesis $p_i \in H$ can be true either by proving it or by assuming it to be true and paying its assumability cost. Hypotheses that can be assumed have assumability costs less than ∞ , we call them “*assumables*”. Consequent hypotheses that are proven through the assumables are called “*provables*”.

Finding the optimal solution for a CBA system is proven to be NP-hard [11] [14]. Previous approaches to CBA can be found in [12] [13] [14]. The only Neural Networks (NN) approach to CBA was introduced in [7], where the authors found the optimal solution of CBA system by transforming it into HORN through an intermediate representation of Penalty Logic (PL).

Finding the LCP in CBA system is equivalent to finding the MAP in BN [11] [14]. Despite their equivalency, it is believed that finding LCP is more efficient than finding MAP and it may be easier to find heuristic for CBA system than finding one for BN [6] [11]. Santos found the necessary and sufficient conditions under which CBA is polynomially solvable [15]. On the other hand, polynomial solvability for finding MAP in BN is not available even with applying restrictions on the graphical representation [4] and even for trying to find an alternative next-best explanation [5].

1.2 High Order Recurrent Networks (HORN)

A recurrent NN is one whose underlying inter-neural connections contain at least one cycle. The Hopfield network is perhaps the most famous recurrent NN [16]. The underlying topology is a graph and each weighted connection is either a binary connection, T_{ij} , that connects two neurons (i, j) or a unary connection I_i which is the bias of a single neuron i . Each neuron is trying to minimize the energy function which is usually composed of two energy functions:

$$E = E^{Obj} + \beta E^{Const} \quad (4)$$

E^{Obj} describes the objective function to be either maximized or minimized while the E^{Const} ensures the feasibility of the optimized solution by enforcing the set of the constraints. β is a problem dependant free parameter to be experimentally tuned. We can think of β as a tradeoff knob between solution optimality and solution feasibility. Depending on the NN order, E can be either quadratic or higher order.

HORN is a recurrent network whose underlying topology is a hypergraph, allowing weighted hyperedges that connect more than two neurons. The degree of the edge is the number of neurons it connects. The order of the network is the highest degree in the topology. In a K^{th} -order HORN a neuron with an activation level u_i and an output V_i is governed by:

$$\frac{du_i}{dt} = \sum_{d=1}^k \sum_{s \in S_d, i \in s} T_s^{(d)} \prod_{j \in s, j \neq i} V_j \quad (5)$$

Where $V_i = g(u_i)$, and g is typically a sigmoidal activation function. k is the order of the network. $T_s^{(d)}$ is the weight of the d^{th} -degree edge connecting neurons $i_1 \dots i_d$. S_d denotes the set of all neuron sequences J_1, \dots, J_d , such that $1 \leq j_{1, \dots, d} \leq n$; where n is the number of neurons and $J_a \neq J_b$ if $a \neq b$ [3]. Each unit minimizes the following K^{th} -order energy function:

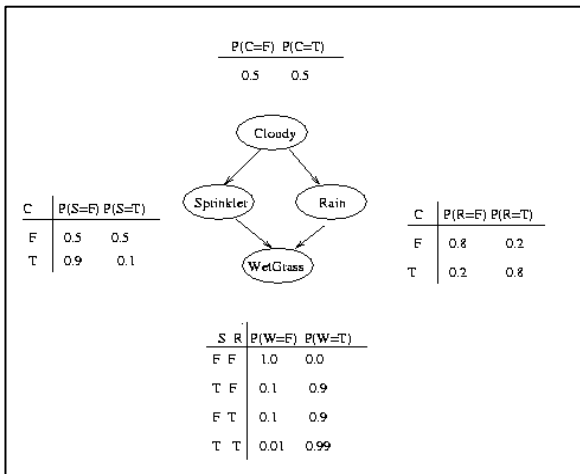
$$\begin{aligned}
 K = & - \sum_{1 \leq i_1 < i_2 < \dots < i_k \leq n} T_{i_1 \dots i_k}^{(k)} V_{i_1} \dots V_{i_k} \\
 & - \sum_{1 \leq i_1 < i_2 < \dots < i_{k-1} \leq n} T_{i_1 \dots i_{k-1}}^{(k-1)} V_{i_1} \dots V_{i_{k-1}} - \dots - \sum_{1 \leq i \leq n} T^{(1)} V_i
 \end{aligned}
 \tag{6}$$

1.3 Related Work

To our knowledge, the only attempt to find MAP using HORN is in [3]. However, this method requires deriving the energy function in two steps: E^{Obj} and E^{Const} ; then, the two functions need to be combined into one function as in (4). This method requires extensive experimentation to set the network parameter β among other parameters [3]. In this work, we will create the full energy function in one step from the CBA system equivalent to the given BN. That will eliminate the need for creating and combining two energy functions and the need for tuning the free parameter β .

2 Transforming BN into CBA

In this section we will follow and analyze the linear time transformation algorithm mentioned in [6] to transform the multiply-connected BN in fig.1 into an equivalent CBA system. This example can be found in Murphy’s BN tutorial [17].



Our objective is to explain why the grass is wet. So we want to reach an assignment A for the random variables such that $P(A|W)$ is maximized. Using Bayesian inference we can see that:

$$P(S = 1 | W = 1) = 0.430$$

$$P(R = 1 | W = 1) = 0.708$$

So, the best explanation for the wet grass is because it is raining rather than because of having the sprinkler on.

Fig. 1. Multiply-connected BN example

The transformation to CBA goes as follows:

1. Transform the CPT of each random variable v to a linear table P_v , tables 1 to 4.

Each line $l \in P_v$ is a hypothesis that the premises of that line are satisfied. $V(l)$ denotes the probability corresponding to line l in the table. The cost of the hypothesis h_l that represents each line is : $c(h_l) = -\log V(l) + Q$ where: $Q = -\log \prod_{v \in V} Q_v$

and $Q_v = \min\{V(l) \mid l \in P_v\}$ So: $Q_C = 0.5, Q_S = 0.1, Q_R = 0.2, Q_W = 0.9$

When calculating Q , we ignore the line $P(W \mid S, R) = 0$ because W is our goal and we are trying to explain why the grass is wet, so $Q = 2.0458$

Table 1. Cloudy , C

Value	$P(C)$	h_l	$c(h_l)$
0	0.5	h_{C1}	2.3468
1	0.5	h_{C2}	2.3468

Table 2. Sprinkler , S

Value		$P(S \mid C)$	h_l	$c(h_l)$
C	S			
0	0	0.5	h_{S1}	2.3468
0	1	0.5	h_{S2}	2.3468
1	0	0.9	h_{S3}	2.0915
1	1	0.1	h_{S4}	3.0458

Table 3. Rain , R

Value		$P(R \mid C)$	h_l	$c(h_l)$
C	R			
0	0	0.8	h_{R1}	2.1427
0	1	0.2	h_{R2}	2.7447
1	0	0.2	h_{R3}	2.7447
1	1	0.8	h_{R4}	2.1427

Table 4. Wet grass, W. The goal

Value			$P(W \mid S, R)$	h_l	$c(h_l)$
S	R	W			
0	0	1	0.0	h_{W1}	∞
1	0	1	0.9	h_{W2}	2.0915
0	1	1	0.9	h_{W3}	2.0915
1	1	1	0.99	h_{W4}	2.0501

2. For every variable $v \in V$, create h_v representing that proposition v is assigned some truth value; $c(h_v) = \infty$. Result is a set of hypothesis $\{h_c, h_s, h_r, h_w\}$

3. For every $v \in V$ and for every $t \in D(v)$, where $D(v)$ is domain of v ,

- o Construct a hypothesis $h_{v,t}$ denoting that proposition v is assigned a value t . Add $h_{v,t}$ to the system hypothesis and assign $c(h_{v,t}) = \infty$;

Result is a set of new hypothesis: $\{h_{C_i}, h_{C_f}, h_{S_i}, h_{S_f}, h_{R_i}, h_{R_f}, h_{W_i}\}$, we ignore h_{W_f} .

- Construct a rule R_{v_i} with $R_{v_i}^A = \{h_{v_i}\}$ and $R_{v_i}^C = \{h_{v_i}\}$

Where R^A refers to the set of R 's antecedents and R^C refers to R 's consequent. Result is this set of rules:

$$\begin{array}{l} R_{C_i} : h_{C_i} \rightarrow h_C, R_{C_f} : h_{C_f} \rightarrow h_C; \quad R_{S_i} : h_{S_i} \rightarrow h_S, R_{S_f} : h_{S_f} \rightarrow h_S \\ R_{R_i} : h_{R_i} \rightarrow h_R, R_{R_f} : h_{R_f} \rightarrow h_R; \quad R_{W_i} : h_{W_i} \rightarrow h_W, R_{W_f} : h_{W_f} \rightarrow h_W \end{array}$$

4. For every $v \in V$ and every $l \in P_v$:

- Construct a rule R_l where: $R_l^A = \{h_l\}$
- For every $\{u \rightarrow t'\} \subseteq l$, where $u \in \pi(v)$ and $t' \in D(u)$, set $R_l^A = R_l^A \cup \{h_{u_t'}\}$
- Let $t \in D(v)$ be the value from v 's domain that satisfies $\{v \rightarrow t\} \subseteq l$, set $R_l^C = \{h_{v_t}\}$. Result is the following sets of rules:

$$\begin{array}{l} R_{C_l1} : h_{C_l1} \rightarrow h_{C_f} \\ R_{R_l1} : h_{R_l1} \wedge h_{C_f} \rightarrow h_{R_f} \\ R_{R_l2} : h_{R_l2} \wedge h_{C_f} \rightarrow h_{R_i} \\ R_{R_l3} : h_{R_l3} \wedge h_{C_i} \rightarrow h_{R_f} \\ R_{R_l4} : h_{R_l4} \wedge h_{C_i} \rightarrow h_{R_i} \end{array} \left| \begin{array}{l} R_{C_l2} : h_{C_l2} \rightarrow h_{C_i} \\ R_{S_l1} : h_{S_l1} \wedge h_{C_f} \rightarrow h_{S_f} \\ R_{S_l2} : h_{S_l2} \wedge h_{C_f} \rightarrow h_{S_i} \\ R_{S_l3} : h_{S_l3} \wedge h_{C_i} \rightarrow h_{S_f} \\ R_{S_l4} : h_{S_l4} \wedge h_{C_i} \rightarrow h_{S_i} \end{array} \right| \begin{array}{l} R_{W_l1} : h_{W_l1} \wedge h_{S_f} \wedge h_{R_f} \rightarrow h_{W_i} \\ R_{W_l2} : h_{W_l2} \wedge h_{S_i} \wedge h_{R_f} \rightarrow h_{W_i} \\ R_{W_l3} : h_{W_l3} \wedge h_{S_f} \wedge h_{R_i} \rightarrow h_{W_i} \\ R_{W_l4} : h_{W_l4} \wedge h_{S_i} \wedge h_{R_i} \rightarrow h_{W_i} \end{array}$$

Finally, the goal set $G = \{h_C, h_W, h_R, h_S, h_{W_i}\}$ and $R_G : h_C \wedge h_W \wedge h_R \wedge h_S \wedge h_{W_i} \rightarrow G$

As discussed above, finding the LCP for this derived CBA system will be the same as finding MAP for BN in fig.1. In other words, the values assigned to CBA variables to reach the LCP are the same values that achieve MAP for the equivalent BN. Section 3 will illustrate how HORN can be used to find LCP for the CBA which will be the same as finding MAP for BN.

2.1 Discussion

The algorithm did a linear time transformation from BN to CBA. However, we have the following comments:

1. There is no analysis in terms of the size ratio between the BN as an input to the CBA system as an output.
2. While the algorithm generates an equivalent CBA system in terms of solution, it might not be the most optimal system in terms of size. We did an experiment where we shrank the R_G to be $h_{W_i} \rightarrow G$ and we obtained the same LCP, but with more computational effort.

3. The algorithm does not remove the redundant CPT entries to save memory space. Also, we do not need to create rules for the entire CPT of the evidence nodes. We only need to create rules for the values observed.
4. It is not clear from the algorithm how we should deal with CPT entries with probability equal to 0. Considering such probabilities will cause all assumables to have costs of ∞ , which means we cannot afford explaining our goal.

3 Finding LCP Using HORN

Here, we will summarize how to solve the CBA system using HORN. The reader is directed to [7] [8] for full details. The process goes as follows:

1. Without loss of generality, we start by processing the CBA system such that all consequents are provable. Also, we make sure that every provable appears only once as a consequent in the system.
2. Given the preprocessed CBA, we reverse the implication direction of all rules to avoid the null antecedent proofs.
3. We transform the CBA into PL pairs. PL is an extension of propositional logic; the reader is directed to [10] for a complete review of PL.
4. We generate the equivalent energy function for all PL formulas using the following characteristic function provided by Pinkas [10]:

$$H_\sigma = \begin{cases} x_i & \text{if } \sigma = x_i \text{ is atomic proposition} \\ 1 - H_{\sigma'} & \text{if } \sigma = \neg\sigma' \\ H_{\sigma_1} \times H_{\sigma_2} & \text{if } \sigma = \sigma_1 \wedge \sigma_2 \\ H_{\sigma_1} + H_{\sigma_2} - H_{\sigma_1} \times H_{\sigma_2} & \text{if } \sigma = \sigma_1 \vee \sigma_2 \end{cases} \quad (7)$$

This function maps every propositional sentence σ into a characteristic algebraic term H_σ that has its maximal points at the truth assignments that satisfy the clause. The equivalent energy function for a given proposition sentence σ is the characteristic function of the sentence negation $H_{\neg\sigma}$. The Energy function for PL pairs $\psi = \bigwedge_i^n \sigma_i$ is defined by (8); this energy function fully specifies the equivalent HORN.

$$E_\Psi = \sum_i^n H_{\neg\sigma} \quad (8)$$

3.1 Deriving Energy Functions for Logical Rules with More Than 3 Variables

In this section we provide derivations of the energy functions for logical rules with more than 3 variables. We start by OR rule; consider $\beta = x_n \rightarrow x_1 \vee x_2 \vee x_3 \vee \dots \vee x_{n-1}$:

$$\begin{aligned} \neg\beta &= x_n \wedge \neg(x_1 \vee x_2 \vee x_3 \vee \dots \vee x_{n-1}) \\ &\equiv x_n \wedge \neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge \dots \wedge x_{n-1} \end{aligned}$$

$$\begin{aligned} \therefore H_{\neg\beta} &= x_n [(1-x_1)(1-x_2)(1-x_3)\dots(1-x_{n-1})] \\ \therefore E_\beta &= x_n - x_n \left(\sum_{k=1}^{n-1} (-1)^{k+1} \left(\sum_{1 \leq i_1 < \dots < i_k \leq n-1} x_{i_1} x_{i_2} \dots x_{i_k} \right) \right) \end{aligned} \quad (9)$$

For AND rule, consider $\beta = x_n \rightarrow x_1 \wedge x_2 \wedge x_3 \wedge \dots \wedge x_{n-1}$

$$\begin{aligned} \neg\beta &\equiv x_n \wedge \neg(x_1 \wedge x_2 \wedge x_3 \wedge \dots \wedge x_{n-1}) \\ &\equiv x_n (1 - x_1 x_2 x_3 \dots x_{n-1}) \end{aligned}$$

$$\therefore H_{\neg\beta} = x_n - x_n x_1 x_2 x_3 \dots x_{n-1}$$

$$\therefore E_\beta = x_n - \prod_{i=1}^n x_i \quad (10)$$

4 Solution Quality and Size Complexity

For the example traced in this manuscript, it is clear that the network reached the LCP and assigned values which give the maximum joint probability for the random variables of the BN in Fig.1. In general, we judge if the network reached the global minima by benchmarking the solution against the results obtained by the popular public domain *lp-solve* engine which solves the CBA system after converting it to the equivalent Linear Programming (LP) form.

The example above showed that HORN solved a problem of size 26-hypothesis, 22-rule. However, previous work [7] [8] showed that HORN constantly found feasible solutions for a CBA system with 300-hypothesis, 900-rule with high difficulty. Problem size is not the only factor which determines the CBA instance difficulty and its search space complexity. Other factors, like solution depth, rules length, and ratio between the number of rules to the total number of hypotheses are taken into consideration when considering a CBA instance difficulty level.

5 Results Summary

Using the previously mentioned transformations, we constructed the energy function which represents the CBA system derived in section 2. Then, we used HORN simulator to minimize this energy function. The LCP was found through the following assignments $\{C \rightarrow T, R \rightarrow T, S \rightarrow F, W \rightarrow T\}$. Total cost of 8.6725 by assuming the following hypotheses $h_c l_2, h_s l_3, h_r l_4$ and $h_w l_3$. This is the same solution we reached using Bayesian inference for BN in fig.1. Table 5 summarizes the results of the HORN which solved this example.

We can also find the LCP by backtracking the rules starting from the goal rule. We only need to calculate towards h_w because all other hypotheses in the goal rule are provables with assumability cost of ∞ . By backtracking rules R_{w_i} 's, it is clear that we cannot use $R_w I_1$ which costs us ∞ , because it explains that the grass is wet while there is neither rain nor sprinkler. That leaves us with rules $R_w I_2$, $R_w I_3$ and $R_w I_4$ with costs : 8.9278, 8.6725 and 9.4884, respectively. That means the best explanation for the observation that the grass is wet is $R_w I_3$ which assumes that the sprinkler is off and there is rain. The LCP assignment of the constructed CBA system is the same assignment for the variables in the BN to achieve MAP.

Table 5. Results summary

R_G	network order	network iterations	cost
$h_c \wedge h_w \wedge h_r \wedge h_s \wedge h_{w_i} \rightarrow G$	9	98489	8.6725
$h_{w_i} \rightarrow G$	9	136128	8.6725

6 Concluding Remarks and Future Work

In this paper we showed how to find MAP in BN using HORN through an intermediate representation of CBA. This method creates the full integrated energy function directly without explicitly setting the objective and constraint functions. We traced and analyzed the only algorithm in literature that transforms BN to CBA. Future work would be to invent new algorithms that transform BN to CBA while taking care of the size ratio between both systems. Finding MAPs for BN with continuous probability distribution will be an interesting follow up for this work. Also, we can study which classes of BN can create polynomially solvable CBA systems.

References

1. McCarthy, J., Hayes, P.J.: Some Philosophical Problems from the Standpoint of Artificial Intelligence. In: Meltzer, B., Michie, D. (eds.) Machine Intelligence, vol. 4, pp. 463–502. Edinburgh University Press (1969)
2. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)
3. Abdelbar, A.M.: Designing high order recurrent networks for Bayesian belief revision. In: Medsker, L.R., Jain, L.C. (eds.) Recurrent Neural Networks: Design and Applications, pp. 77–98. CRC Press, Boca Raton (1999)
4. Shimony, S.E.: Finding MAPs for belief networks is NP-hard. Artificial Intelligence 68, 399–410 (1994)
5. Abdelbar, A.M., Hedetniemi, S.M.: Approximating MAPs for belief networks is NP-hard and other theorems. Artificial Intelligence 102, 21–38 (1998)

6. Abdelbar, A.M.: An algorithm for finding MAPs for belief networks through cost-based abduction. *Artificial Intelligence* 104, 331–338 (1998)
7. Abdelbar, A.M., Andrews, E.A.M., Wunsch II, D.C.: Abductive Reasoning with Recurrent Neural Networks. *Neural Networks* 16(5-6), 665–673 (2003)
8. Abdelbar, A.M., El-Hemely, M.A., Andrews, E.A.M., Wunsch II, D.C.: Recurrent Neural Networks with Backtrack-Points and Negative Reinforcement Applied to Cost-Based Abduction. *Neural Networks* 18(5-6), 755–764 (2005)
9. Charniak, E., Shimony, S.E.: Probabilistic semantics for cost-based abduction. In: AAAI National Conference on Artificial Intelligence, pp. 106–111. AAAI, Boston (1990)
10. Pinkas, G.: Reasoning, nonmonotonicity and learning in connectionist networks that capture propositional knowledge. *Artificial Intelligence* 77, 203–347 (1995)
11. Charniak, E., Shimony, S.E.: Cost-based abduction and MAP explanation. *Artificial Intelligence* 66, 345–374 (1994)
12. Den, Y.: Generalized Chart Algorithm: An Efficient Procedure for Cost-Based Abduction. In: 32nd annual Meeting of the Association for Computational Linguistics, pp. 218–225. New Mexico Association for Computational Linguistics, Las Cruces (1994)
13. Ishizuka, M., Matsuo, Y.: SL Method for Computing a Nearoptimal Solution Using Linear and Non-Linear Programming in Cost-Based Hypothetical Reasoning. *Knowledge-based systems* 15, 369–376 (2002)
14. Santos, E.: A Linear constraint satisfaction approach to Cost-Based Abduction. *Artificial Intelligence* 65(1), 1–27 (1994)
15. Santos, E.J., Santos, E.S.: Polynomial Solvability of Cost-Based Abduction. *Artificial Intelligence* 86, 157–170 (1996)
16. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. *National Academy of Science* 79, 2554–2558 (1982)
17. Murphy, K.: A Brief Introduction to Graphical Models and Bayesian Networks, <http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html>

A Study on Bayesian Learning of One-Dimensional Linear Dynamical Systems

Takuto Naito¹ and Keisuke Yamazaki²

¹ Department of Computational Intelligence and Systems Science,
Tokyo Institute of Technology,
R2-5, 4259 Nagatsuta, Midori-Ku, Yokohama, Kanagawa, Japan
`naitaku@cs.pi.titech.ac.jp`

² Precision and Intelligence Laboratory,
Tokyo Institute of Technology
`k-yam@pi.titech.ac.jp`

Abstract. Linear dynamical systems are widely used in such fields as system control and time-dependent data analysis. Such a system can be regarded as a statistical parametric model, where the coefficients of the state space equations are unknown and given as parameters. The properties of parameter learning have not yet been established, in spite of a wide range of applications. Therefore, this paper investigates the system from the viewpoint of learning theory. It is revealed that the system has singularities in the parameter space. The generalization error measured by the prediction accuracy for unseen data sequences is reduced, due to the presence of these singularities.

Keywords: Kalman Filter, Bayesian Learning, Time-Series Data Analysis.

1 Introduction

Linear dynamical systems are widely used for modeling practical complex systems with hidden variables such as object tracking in image processing [1], and position detection in car navigation systems [2]. The system is described via state space equations containing both observable and hidden variables. The Kalman filter [3] is an algorithm to estimate the hidden variables from coefficients given preliminarily .

It is important to be able to estimate coefficients using the observable data when the coefficients are unknown. The system is regarded as a parametric learning model, in which the coefficients correspond to parameters. As seen in Section 2 the system is expressed as a generative probability model of the data because the process and observation noises are taken into account.

Parametric models generally fall into two types, *regular* and *singular*. If the relation between the parameter and the expressed probability function is one-to-one, a model is referred to as regular. Otherwise, it is singular. Therefore, a singular model has a set of parameters indicating the same function, in which

there are singularities. Because of the singularities, conventional analysis is not applicable; model selection criteria for regular models such as AIC [4] and BIC [5] are inappropriate. An algebraic geometrical method has been developed for Bayesian learning to reveal the asymptotic generalization error and the marginal likelihood for several singular models [6]. According to its application to several models, the presence of singularities results in unique properties of the learning process [7,8].

In spite of a wide range of applications, properties of a linear dynamical system are still unknown in terms of a learning model. Therefore, the present paper investigates such a system both theoretically and experimentally. We confirm that the system is a singular model and analyze the Bayesian generalization error based on the algebraic geometrical method. Here, the error is defined as the prediction accuracy for unseen time-sequence data. This *prediction* is different from that of the conventional Kalman situation in which the primary concern is the set of hidden variables rather than the observable sequences. Nevertheless, our analysis can also provide an insight into hidden variable estimation.

The remainder of the paper is organized as follows. Section 2 formulates the system. Section 3 introduces Bayesian learning and summarizes the algebraic geometrical method. Section 4 contains our main contributions, deriving a theoretical upper bound of the generalization error and showing experimental results for the error. Section 5 contains a discussion and our conclusions.

2 Linear Dynamical Systems

Linear dynamical systems can be described by state space models with hidden state variables:

$$z_{t+1} = Az_t + Dw_t, \quad (1)$$

$$x_t = Cz_t + v_t, \quad (2)$$

where $z_t \in \mathbf{R}^q$ is the hidden state vector at time t , $x_t \in \mathbf{R}^p$ is an output vector, $w_t \in \mathbf{R}^q$ and $v_t \in \mathbf{R}^p$ are process and observation noises, respectively. These noises are assumed follow a standard normal distribution. $A \in \mathbf{R}^{q \times q}$ is the state matrix, $C \in \mathbf{R}^{p \times q}$ is the output matrix and the elements of $D \in \mathbf{R}^{q \times q}$ are the coefficients of the process noise.

The Kalman filter is known as an efficient recursive filter that estimates hidden states from a series of outputs. In what follows, the notations $\hat{z}_{n|m}$ and $P_{n|m}$ represent the estimates of z at time n and its error covariance matrix, respectively, when observations from $t = 1$ to $t = m$ are given. The Kalman filter has two phases: **Predict** and **Update**. The algorithms are described as follows:

Predict

$$\hat{z}_{t|t-1} = A\hat{z}_{t-1|t-1} \quad (3)$$

$$P_{t|t-1} = AP_{t-1|t-1}A^\top + DD^\top \quad (4)$$

Update

$$K_t = P_{t|t-1}C^\top (I + CP_{t|t-1}C^\top)^{-1} \tag{5}$$

$$\hat{z}_{t|t} = \hat{z}_{t|t-1} + K_t (x_t - C\hat{z}_{t|t-1}) \tag{6}$$

$$P_{t|t} = (I - K_tC) P_{t|t-1} \tag{7}$$

where I is a unit matrix and K_t is called the Kalman gain. Firstly, the current state z_t is estimated as $\hat{z}_{t|t-1}$ from the estimated state of the previous time $t - 1$ (Eq.3). Then, a more refined value for $\hat{z}_{t|t}$ is calculated on the basis of $\hat{z}_{t|t-1}$ after an observation x_t is provided (Eq.6).

From the viewpoint of machine learning, a linear dynamical system can be regarded as a learning model whose parameters are A, C, D and z_1 . The variable z_1 indicates the initial state. Let $X = (x_1, x_2, \dots, x_T) \in \mathbf{R}^{p \times T}$ be the vector of observations. The probability $p(X|w)$, where the parameters $w = (A, C, D, z_1)$, can be calculated as follows:

$$p(X|w) = p(x_1|w) \prod_{t=2}^T p(x_t|x_1, \dots, x_{t-1}, w). \tag{8}$$

Using the hidden state z_t ,

$$p(x_t|x_1, \dots, x_{t-1}, w) = \int p(x_t|z_t, w)p(z_t|x_1, \dots, x_{t-1}, w)dz_t. \tag{9}$$

Let $\mathcal{N}(\cdot|\mu, \Sigma)$ be a multivariate normal distribution with mean μ and covariance matrix Σ . By the definition of a linear dynamical system (Eq.2) and the derivation of the Kalman filter,

$$p(x_t|z_t, w) = \mathcal{N}(x_t|Cz_t, I), \tag{10}$$

$$p(z_t|x_1, \dots, x_{t-1}, w) = \mathcal{N}(z_t|\hat{z}_{t|t-1}, P_{t|t-1}). \tag{11}$$

Therefore, $p(x_t|x_1, \dots, x_{t-1}, w)$ is also a normal distribution described by

$$p(x_t|x_1, \dots, x_{t-1}, w) = \mathcal{N}(x_t|C\hat{z}_{t|t-1}, I + CP_{t|t-1}C^\top). \tag{12}$$

Eq.8 can be expressed as

$$p(X|w) = \prod_{t=1}^T \mathcal{N}(x_t|C\hat{z}_{t|t-1}, I + CP_{t|t-1}C^\top). \tag{13}$$

where we define $\hat{z}_{1|0} = z_1$ and $P_{1|0} = 0$.

Let $X^n = (X_1, X_2, \dots, X_n)$ be a set of i.i.d. training samples. Each X_i is a time sequence defined by $X_i = (x_1^i, x_2^i, \dots, x_T^i)$. The likelihood of the parameter $w = (A, C, D, z_1)$ can be calculated as

$$L(w) = \prod_{i=1}^n p(X_i|w) = \prod_{i=1}^n \prod_{t=1}^T \mathcal{N}(x_t^i|C\hat{z}_{t|t-1}^i, I + CP_{t|t-1}^iC^\top) \tag{14}$$

where $\hat{z}_{t|t-1}^i$ and $P_{t|t-1}^i$ are evaluated using the Kalman filter.

3 Bayesian Learning and the Generalization Error

This section describes Bayesian learning for time series data and the theoretical analysis of the generalization error.

Let $X^n = (X_1, X_2, \dots, X_n)$ be a set of training samples taken independently and identically from the true distribution $q(X)$, where n is the number of training samples. Each X_i ($i = 1, \dots, n$) is a sequence whose length is T , i.e. $X_i = (x_1^i, \dots, x_t^i, \dots, x_T^i)$. Note that the sequence data X^n are taken as i.i.d. whereas each sequence X_i is not. Let $p(X|w)$ be a learning model, and $\varphi(w)$ be an a priori probability distribution. The a posteriori probability distribution is defined by

$$p(w|X^n) = \frac{1}{Z(X^n)} \varphi(w) \prod_{i=1}^n p(X_i|w) \quad (15)$$

where $Z(X^n)$ is a normalizing constant. The Bayesian predictive distribution is defined by

$$p(X|X^n) = \int p(X|w)p(w|X^n)dw. \quad (16)$$

The Bayesian generalization error $G(n)$ is defined by

$$G(n) = E_{X^n} \left[\int q(X) \log \frac{q(X)}{p(X|X^n)} dX \right], \quad (17)$$

which is the average Kullback information from the true distribution to the predictive distribution.

The remainder of this section summarizes the algebraic geometrical method for deriving the asymptotic form of the error [6]. Let $H(w)$ be the Kullback information from the true distribution $q(X)$ to the learner $p(X|w)$,

$$H(w) = \int q(X) \log \frac{q(X)}{p(X|w)} dX. \quad (18)$$

The function $\zeta(z)$ of one complex variable z , defined by

$$\zeta(z) = \int H(w)^z \varphi(w) dw, \quad (19)$$

is referred to as the zeta function. It is known that this zeta function is holomorphic in the region $\text{Re}(z) > 0$, and can be analytically continued to the meromorphic function on the entire complex plane. Then the poles are all real, negative and rational numbers. Let $0 > -\lambda_1 > -\lambda_2 > \dots$ be a sequence of poles, and m_1, m_2, \dots be the respective orders. The asymptotic form of the generalization error is expressed as

$$G(n) = \frac{\lambda_1}{n} - \frac{m_1 - 1}{n \log n} + o\left(\frac{1}{n \log n}\right) \quad (20)$$

for $n \rightarrow \infty$. In many cases, it is not straightforward to find the largest pole $-\lambda_1$ and its order m_1 [7]. When a pole $z = -\lambda$ and its order m have been calculated, an upper bound is derived as

$$G(n) \leq \frac{\lambda}{n} - \frac{m-1}{n \log n} + o\left(\frac{1}{n \log n}\right). \quad (21)$$

4 Analysis of the Generalization Error

This section analyzes the Bayesian generalization error for linear dynamical systems. In order to investigate the effect of redundant hidden states, we study an essential case, in which the learning model has a hidden variable and the true model generates i.i.d. sequences. This is the simplest setting for singularities to exist in the parameter space because the i.i.d. model can be regarded as a model with no hidden states. For simplicity, we assume that the output vector is one dimensional, where z_t , x_t , A , C , and D are all scalar. Moreover, we assume that the first hidden state is fixed as $z_1 = 0$. Formally, the learning model is defined as

$$z_{t+1} = az_t + dw_t, \quad (22)$$

$$x_t = cz_t + v_t, \quad (23)$$

where $z_t, x_t \in \mathbf{R}^1$ and w_t and v_t are distributed from $\mathcal{N}(\cdot|0, 1)$. The parameter is expressed as $w = (a, c, d)$. The true model is a one-dimensional normal distribution $\mathcal{N}(x_t|0, 1)$ for all t , i.e. $x_t = v_t$. Following Eq. [13], the true model is given by

$$q(X) = \prod_{t=1}^T \mathcal{N}(x_t|0, 1). \quad (24)$$

4.1 Theoretical Analysis

Based on the algebraic geometrical method, the error has the following bound:

Theorem 1. *When the true model and a learning model are defined by Eq. [24] and Eqs [22-23], respectively, the Bayesian generalization error is bounded above as follows:*

$$G(n) \leq \frac{1}{2n} - \frac{1}{n \log n} + o\left(\frac{1}{n \log n}\right), \quad (25)$$

where $z_1 = 0$ and the training sample size n is sufficiently large.

Sketch of Proof: Because the parameter set $\{c = 0\}$ attains $p(X|w) = q(X)$, there is a function $f_c(w)$ such that $H(w) = c^2 f_c(w)$. The set $\{d = 0\}$ ensures the same property for $H(w)$. Thus, there is a polynomial $f(w)$ such that $H(w) = c^2 d^2 f(w)$. We can find a limited support W of the parameter space, such that

$H(w) \leq Cc^2d^2$. Here C is a positive constant. Considering the following zeta function

$$\zeta_1(z) = \int_W \{Cc^2d^2\}^z dw, \tag{26}$$

the pole $z = -\mu$ is a lower bound of $z = -\lambda_1$ [6]. We can find a pole $\mu = 1/2$ and its order $m = 2$. Combining with Eq. 21, we derive the following leading terms for the bound,

$$\frac{1}{2n} - \frac{1}{n \log n}, \tag{27}$$

which completes the proof.

End of Proof

If the initial state is unknown and is regarded as a parameter such as $w = (a, c, d, z_1)$, we can extend Theorem 1 as follows.

Corollary 1. *Under the same setting as Theorem 1, the error has an upper bound*

$$G(n) \leq \frac{1}{2n} + o\left(\frac{1}{n}\right). \tag{28}$$

We omit the proof for lack of space.

4.2 Experimental Results

We experimentally evaluate whether the bound is valid when finite training data are given. Sampling from the a posteriori distribution, the predictive distribution is given by

$$p(X|X^n) \simeq \frac{1}{M} \sum_{j=1}^M p(X|w_j), \tag{29}$$

where (w_1, \dots, w_M) are sampled from $p(w|X^n)$. We use the Markov chain Monte Carlo (MCMC) method for the sampling technique [9]. The generalization error is approximated by

$$G(n) \simeq E_{X^n} \left[\frac{1}{N} \sum_{i=1}^N \log \frac{q(X_i)}{p(X_i|X^n)} \right]. \tag{30}$$

The experimental settings are as follows. The length of the time sequence is $T = 10$. The number of test data sequences is $N = 1,000$. The number in the MCMC sample is $M = 500$. We obtain the expectation $E_{X^n}[\cdot]$ over 100 sets of training data. The a priori distribution is a normal distribution for a, c and d .

Figure 1(a) describes an example of sampling from the a posteriori distribution in the parameter space (a, c, d) . The vertical and horizontal planes indicate

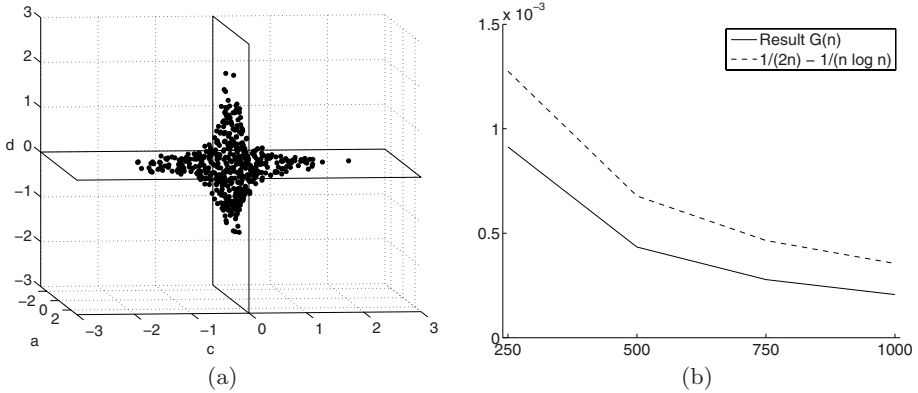


Fig. 1. An example of the a posteriori distribution and the generalization error

$\{c = 0\}$ and $\{d = 0\}$, respectively. The points are located around the subspace $\{c = 0\} \cup \{d = 0\}$, for which the parameters express the true model.

Figure 1(b) summarizes the error values corresponding to $n = 250, 500, 750$ and $1,000$. The horizontal and vertical axes describe the number of training sequences and the error value, respectively. The heavy line depicts experimental values for $G(n)$. The dotted line is the upper bound of Theorem 1. The upper bound is valid as seen in the graph.

5 Discussions and Conclusions

First, let us discuss the upper bound of the generalization error. In the regular case, the error has the following asymptotic form,

$$G(n) = \frac{\dim w}{2n} + o\left(\frac{1}{n \log n}\right), \tag{31}$$

which means that $\lambda_1 = \dim w/2$ and $m_1 = 1$. Note that even a singular model has this asymptotic form if the true and learning models have the same dimension of the hidden state vector. The asymptotic form indicates that the cost to fit all parameters determines the error as the dimension $\dim w$ appears. Comparing Theorem 1 with the regular case, we can derive the result that the error is much smaller, i.e.

$$G(n) \leq \frac{1}{2n} - \frac{1}{n \log n} + o\left(\frac{1}{n \log n}\right) < \frac{3}{2n} + o\left(\frac{1}{n \log n}\right), \tag{32}$$

which confirms that the fitting cost for redundant parameters is not strongly reflected in the error.

Thus far, we have focused on prediction of the unseen observable data sequence X . Next, we consider estimation of the hidden states z_t . According to the a

posteriori distribution, there are two regions for the optimal parameters; one is around $c = 0$ and the other is around $d = 0$. They imply completely different behaviors of the hidden state. The former, $c = 0$, indicates that a and d can take any value, by which $q(X) = p(X|w)$. Thus, there are no constraints on the movement of the hidden state. By taking into account $z_1 = 0$, the latter, $d = 0$, contrarily implies that there is no movement because $z_t = 0$ for all times t . If several hidden variables in the true model stop moving due to disorder in a practical situation, the desired estimation is $d = 0$. However, $c = 0$ can also be an estimated result; these variables move on the basis of arbitrarily-estimated a and d . This adverse estimation can occur along any dimension of the hidden state vector. Therefore, detection of hidden variable size is an essential problem to solve.

Finally, we state our conclusions. The present paper establishes that linear dynamical systems are singular models. The singularities ensure that the upper bound of the Bayesian generalization error is small. The experimental results indicate that the bound is valid. Moreover, the a posteriori distribution implies that estimation of hidden states cannot be appropriate if there are redundant hidden variables.

Acknowledgment

This research was partially supported by the Ministry of Education, Science, Sports and Culture in Japan, Grant-in-Aid for Scientific Research 18079007.

References

1. Funk, N.: A study of the Kalman filter applied to visual tracking. Technical Report Project for CMPUT 652, University of Alberta (2003)
2. Obradovic, D., Lenz, H., Schupfner, M.: Sensor fusion in siemens car navigation system. In: Proc. of MLSP 2004, pp. 655–664 (2004)
3. Kalman, R.E.: A new approach to linear filtering and prediction problems. *J. Basic Engineering* 82, 35–45 (1960)
4. Akaike, H.: A new look at the statistical model identification. *IEEE Trans. on Automatic Control* 19, 716–723 (1974)
5. Schwarz, G.: Estimating the dimension of a model. *Annals of Statistics* 6 (2), 461–464 (1978)
6. Watanabe, S.: Algebraic analysis for nonidentifiable learning machines. *Neural Computation* 13(4), 899–933 (2001)
7. Aoyagi, M., Watanabe, S.: Stochastic complexities of reduced rank regression in Bayesian estimation. *Neural Networks* 18, 924–933 (2005)
8. Yamazaki, K., Watanabe, S.: Algebraic geometry and stochastic complexity of hidden Markov models. *Neurocomputing* 69(1-3), 62–84 (2005)
9. Andrieu, C., de Freitas, N., Doucet, A., Jordan, M.I.: An introduction to MCMC for machine learning. *Machine Learning* 50(1-2), 5–43 (2003)

Decoding Characteristics of D/A Converters Based on Spiking Neurons

Masao Takiguchi and Toshimichi Saito

Hosei University, Koganei, Tokyo, 184-8584 Japan

Abstract. This paper studies spike-based D/A converters and effects of a control parameter on the worst error for the encoding. First, we introduce spike-based A/D converters and analyze their dynamics through 1-D linear map. Next, we present spike-based D/A converters whose architectures is based on inverse operation of the A/D converters. We consider effects of a parameter for decoding function. A simple circuit model of the D/A converter is also presented.

1 Introduction

Spiking neurons (ab. SKNs) are known as simple artificial neuron models [1]-[4]. Repeating integrate-and-fire behavior between base and threshold signals, the SKNs can generate a spike-train [5]. Depending on the shape the base signal, the SKNs can output a variety of spike-trains and can exhibit interesting bifurcation phenomena. Analysis of the bifurcation phenomena is an important nonlinear problem. If we fix a parameter of the base signal, the SKNs have functions as A/D converter (ab. ADC) [6]-[7]. For the ADC, an analog input is applied as an initial value and the spike-train is transformed into a digital output. The architectures of D/A converter (ab. DAC) are based on inverse operation of the ADC. For the DAC, digital input is the inverse sequence of digital output of ADC and an analog output is given as phase of final spike within a given limited time [8]-[9].

This paper studies the spike-based DAC and effects of a control parameter on the decoding function. First, as a preparation, we introduce the ADCs based on SKNs. It has periodic base signal and outputs spike-trains governed by a spike-position map. The map is one dimensional piecewise linear and we can analyze ADC dynamics precisely. Next, we present the DACs based on SKNs. We introduce a parameter that changes a shape of the base signal and the digital output of the ADC. We analyze effects of the parameter on the decoding function. This parameter corresponds to parameter miss match in realistic systems and cause error in the decoding. Such analysis is important to consider basic DAC performance, however, the analysis has not been sufficient in our previous works. We also present a simple circuit model of SKN for the DAC. The circuit consists of a current source, a capacitor and a firing switch. The digital input switches the base signal and the switching base is a key to realize the SKN-based DAC.

ADCs and DACs are crucial systems in order to communicate between real analog world and digital signal processing system. As compared with existing

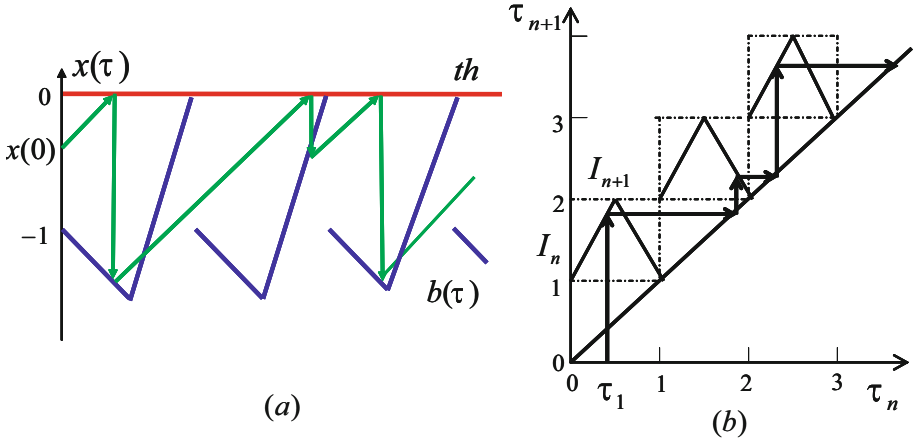


Fig. 1. Basic maps for SKN-based ADC. (a) SKN dynamics, (b) Spike-position map.

architectures, SKN-based ADC and DAC have some advantages such as consuming lower power, avoiding divergence of state variable [6] [7]. This paper may contribute to bridge between neuro dynamics and signal processing.

2 A/D Converters

ADCs convert a constant analog input $x \in I \equiv (0, 1]$ to a digital output sequence $Y \equiv (Y_1, Y_2, \dots, Y_l), Y_n \in \{0, 1\}$ where l is code-length. Here we introduce a SKN-based ADC [2]. As shown in Fig. 1(a), the SKN dynamics is described by Eq. (1).

$$\begin{cases} \frac{dx}{d\tau} = 1, & y = 0, & \text{for } x(\tau) < 0 \\ x(\tau^+) = b(\tau^+), & y(\tau^+) = 1, & \text{if } x(\tau) = 0 \end{cases} \quad (1)$$

where τ and x are normalized time and state variable, respectively. $b(\tau)$ is a base signal with period 1. The state x rises to a threshold 0. When x reaches 0, x jumps to the base $b(\tau)$ instantaneously and the SKN outputs a spike $y = 1$. Adjusting the shape of $b(\tau)$, the SKN can output a variety of spike-trains. In this section, we use the following shape:

$$b(\tau) = \begin{cases} (1 - \frac{1}{\alpha})\tau - 1, & \text{for } 0 \leq \tau < \alpha, \\ \frac{2-\alpha}{1-\alpha}\tau - \frac{1}{1-\alpha} - 1, & \text{for } \alpha \leq \tau < 1, \end{cases} \quad b(\tau + 1) = b(\tau). \quad (2)$$

where α changes a slope of the base $b(\tau)$.

Let τ_n denote the n -th spike-position. Let an initial pulse position τ_1 be in $[0, 1) \equiv I_1$. Since present spike-position determines the next position, we can define a spike position map (Fig. 1(b)):

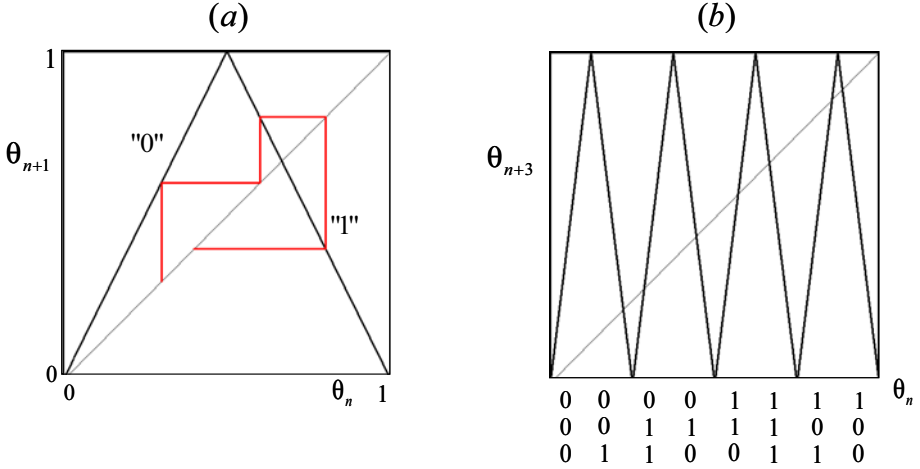


Fig. 2. Key maps for Gray ADC ($l = 3, \alpha = 0.5, \theta_1 = 0.3$). (a) spike-phase map, (b) encoding characteristics. θ_{n+3} denotes the 3 fold compositions of θ_n .

$$\tau_{n+1} = \tau_n - b(\tau_n), \quad \tau_{n+1} = \begin{cases} \frac{1}{\alpha}\tau_n + 1, & \text{for } 0 \leq \tau_n < \alpha \\ \frac{1}{1-\alpha}(1 - \tau_n) + 1, & \text{for } \alpha \leq \tau_n < 1 \end{cases} \quad (3)$$

Let us introduce subintervals of time $I_n \equiv [n - 1, n)$. This map is piecewise affine and transforms I_n onto I_{n+1} . Let θ_n denote phase of the n -th spike position: $\theta_n = \tau_n \bmod 1$. The phase sequence is described by the following spike-phase map (Fig. 2(a)).

$$\theta_{n+1} = \begin{cases} \frac{1}{\alpha}\theta_n, & \text{for } 0 \leq \theta_n < \alpha \\ \frac{1}{1-\alpha}(1 - \theta_n), & \text{for } \alpha \leq \theta_n < 1 \end{cases} \quad (4)$$

This map corresponds to usual return map. In order to characterize spike-trains, we introduce spike-phase modulation (ab. SPM).

$$Y_n = \begin{cases} 0, & \text{for } 0 \leq \theta_n < \alpha \\ 1, & \text{for } \alpha \leq \theta_n < 1 \end{cases} \quad (5)$$

The spike-phase map functions as ADC by the SPM. When $\alpha = 0.5$, it gives gray encoding as suggest in Fig. 2. If $\alpha \neq 0.5$, it changes the code sequence by the initial value in ADC (Fig. 3). α changes the shape of return map and the appearance probability of the Gray code.

3 D/A Converters

The DAC is required to realize inverse operation of the ADC. That is, the DAC converts a digital input $Y' \equiv (Y'_1, Y'_2, \dots, Y'_l), Y'_l \in \{0, 1\}$ into an analog output

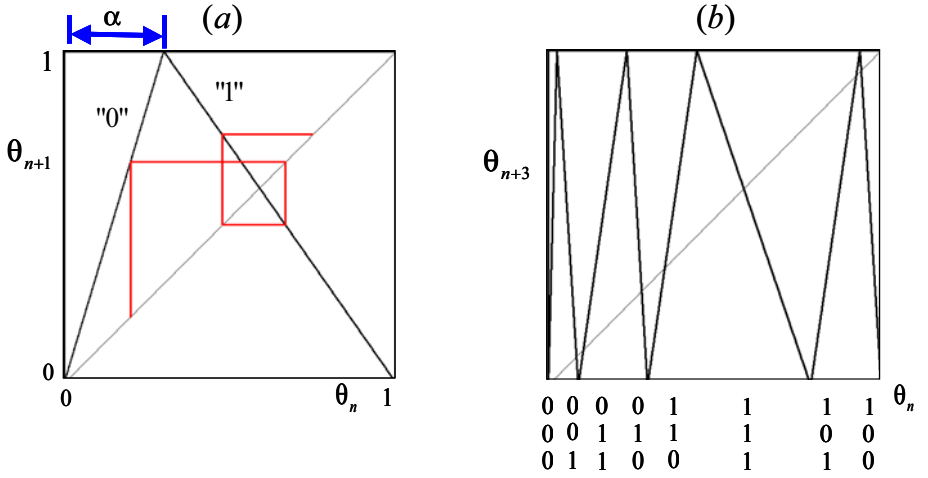


Fig. 3. Skew tent maps for ADC ($l = 3, \alpha = 0.3, \theta_1 = 0.2$). (a) spike-phase map, (b) encoding characteristics.

x' . where Y' is the inverse sequence of Y with code length l : $(Y'_1, Y'_2, \dots, Y'_n) = (Y_n, Y_{n-1}, \dots, Y_1)$. We present the SKN-based DAC in this section. As shown in Fig. 4(a), the SKN dynamics is described by Eqs. (6) and (7).

$$\begin{cases} \frac{dx}{d\tau} = 1, y = 0, & \text{for } x(\tau) < 0 \\ x(\tau^+) = b(\tau^+), y(\tau^+) = 1, & \text{if } x(\tau) = 0 \end{cases} \quad (6)$$

$$b(\tau) = \begin{cases} (1 - \alpha)\tau - 1, & \text{if } Y'_n = 0, \\ (2 - \alpha)\tau - 2, & \text{if } Y'_n = 1, \end{cases} \quad b(\tau + 1) = b(\tau). \quad (7)$$

It should be noted that the digital input (Y'_1, \dots, Y'_l) switches the base signal. In a likewise manner as the ADC, we can derive the spike-position map. Let τ'_n denote the n -th spike-position and let an initial pulse position τ'_1 be in $[0, 1)$. Since present spike-position determines the next position τ'_{n+1} , we can define a spike-position map (Fig. 4(b)):

$$\tau'_{n+1} = \tau'_n - b(\tau'_n), \quad \tau'_{n+1} = \begin{cases} \alpha\tau'_n + 1, & \text{if } Y'_n = 0 \\ -(1 - \alpha)\tau'_n + 2, & \text{if } Y'_n = 1 \end{cases} \quad (8)$$

Let θ_n denote phase of the n -th spike position: $\theta_n = \tau_n \bmod 1$. The phase sequence is described by the following spike-phase map.

$$\theta'_{n+1} = \begin{cases} \alpha\theta'_n, & \text{if } Y'_n = 0 \\ -(1 - \alpha)\theta'_n + 1, & \text{if } Y'_n = 1 \end{cases} \quad (9)$$

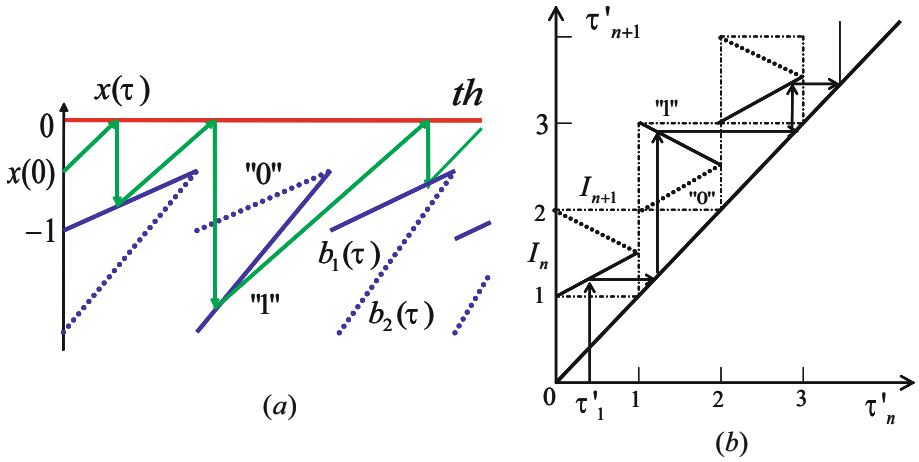


Fig. 4. Basic maps for SKN-based DAC. (a) SKN dynamics, (b) spike-position map.

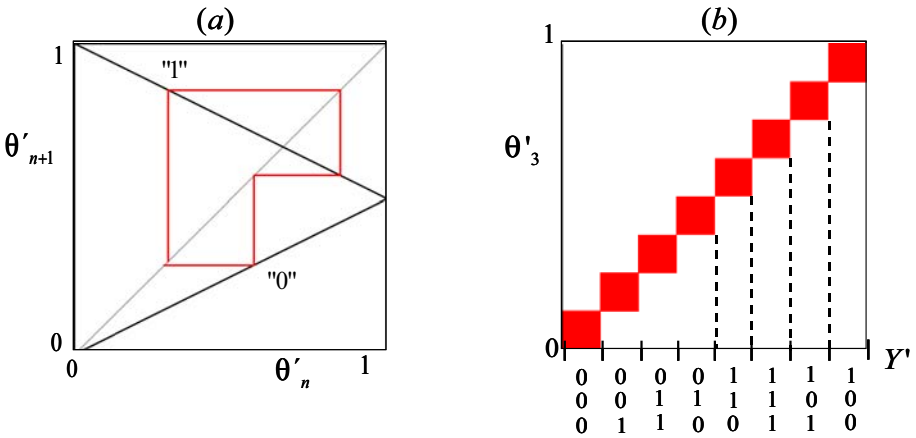


Fig. 5. Key maps for Gray ADC ($l = 3, \alpha = 0.5, Y'_n = (1, 1, 0), \theta'_1 = 0.3$). (a) spike-phase map, (b) decoding characteristics.

The spike-phase map works as DAC based on the SPM. It is the inverse map of ADC. Note that the decoding sequence $\{\theta_n\}$ is determined by the digital inputs Y'_n , and the initial value θ'_1 is optional. When $\alpha = 0.5$, it gives gray decoding as suggest in Fig. 5(a). α changes a shape of map and an analog output. Fig. 5(b) and Fig. 6(b) show spike-phase θ'_3 for all the initial value θ'_1 in $(0, 1]$. θ'_3 is the analog output for $l = 3$. Note that range of the DAC in Fig. 6(b) is consistent with the domain of the ADC in Fig. 3(b).

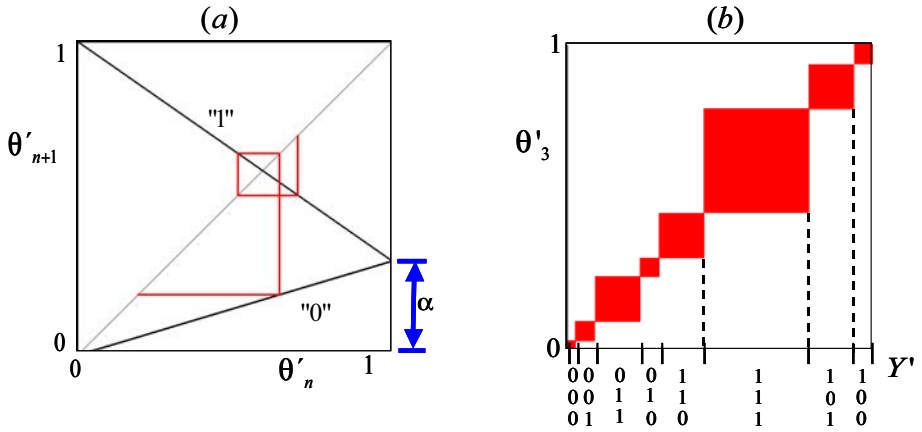


Fig. 6. Skew tent maps for DAC ($l = 3, \alpha = 0.3, Y'_n = (1, 1, 0), \theta'_1 = 0.7$). (a) spike-phase map, (b) decoding characteristics.

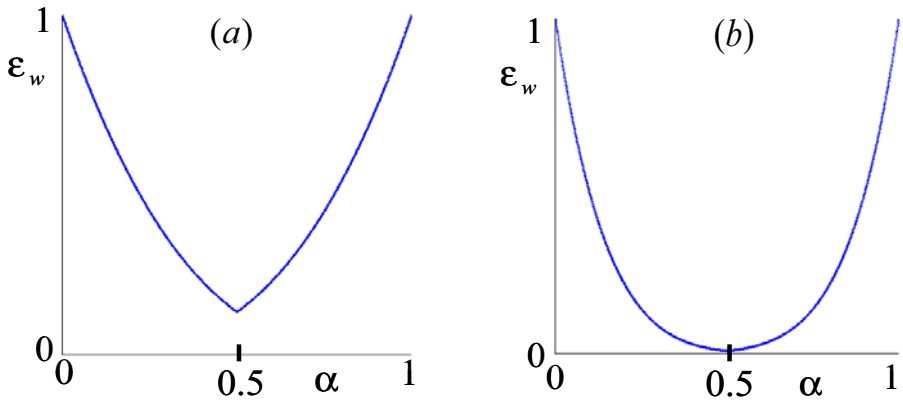


Fig. 7. The worst error by α (a) $l = 3$, (b) $l = 7$

Let the worst error ε_w in decoding. ε_w is the worst error in all decoding and calculated by Eq. (10):

$$\varepsilon_w = \max_{\theta_1 \in I_1} |\theta'_l - \theta_1| \quad (10)$$

where θ_1 is the initial value in ADC. The worst error ε_w is defined by

$$\varepsilon_w \leq \alpha^r (1 - \alpha)^{l-r} \quad (11)$$

where r is the frequency of code 0 in the code sequence. Fig. 7 shows the worst error in changing α . As l increases, ε_w tends to be small. $\alpha = 0.5$ gives the smallest ε_w .

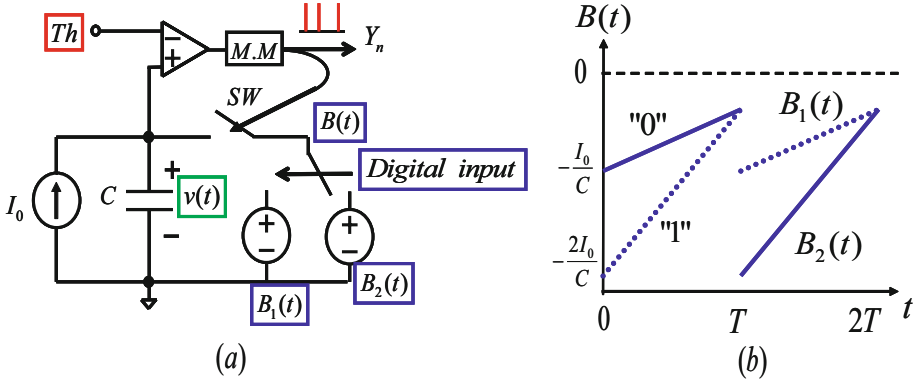


Fig. 8. (a) SKN-based DAC circuit, (b) base signals

4 Circuit Model

Fig. 8 shows the circuit model of the SKN-based DAC. Integrating the current I_0 the capacitor voltage $v(t)$ rises. When it reaches a threshold $Th(t)$, the comparator triggers the monostable multivibrator MM to output a pulse Y . The pulse Y closes the switch SW and v is reset to the base $B(t)$, that is controlled by digital input Y'_n . Repeating this integrate-and-fire behavior this circuit outputs a spike-train $Y(t)$. For simplicity the circuit operation is assumed to be ideal: the current source is lossless and the switching is instantaneous without time delay. The circuit dynamics is described by Eq. (12), (13) and (14).

$$\begin{cases} C \frac{dv}{dt} = I_0, Y(t) = -E, & \text{for } v < Th \\ v(t^+) = B(t^+), Y(t^+) = E, & \text{if } v(t) = Th \end{cases} \quad (12)$$

$$B(t) = \begin{cases} B_1(t) = \frac{I_0}{C}(1 - \alpha)t - \frac{I_0 T}{C}, & \text{if } Y'_n = 0, \\ B_2(t) = \frac{I_0}{C}(2 - \alpha)t - \frac{2I_0 T}{C}, & \text{if } Y'_n = 1, \end{cases} \quad \text{for } 0 < t < T. \quad (13)$$

$$B(t + T) = B(t). \quad (14)$$

They are transformed into Eq. (7) using the following dimensionless parameters and variables.

$$\begin{cases} \tau = \frac{t}{T}, x = \frac{C}{I_0 T} v, th(\tau) = 0, y = \frac{Y+E}{2E}, \\ b(\tau) = \frac{C}{I_0 T} B(t). \end{cases} \quad (15)$$

Adjusting shapes of $B(t)$ and $Th(t)$, this circuit can also realize ADC: it is a reconfigurable circuit.

5 Conclusions

We studied spike-based DAC and effects of a control parameter on the worst error for the encoding. We present spike-based ADC and DAC. The architecture of DAC is based on inverse operation of the ADC. We consider effects of a parameter for decoding function. A simple circuit model of the DAC is also presented. In order to develop these results, we have many future problems including detailed error analysis for circuit parameters, generalization of encoding function and the hardware experiments.

References

1. Perez, R., Glass, L.: Bistability, period doubling bifurcations and chaos in a periodically forced oscillator. *Phys. Lett.* 90A(9), 441–443 (1982)
2. Torikai, H., Saito, T., Schwarz, W.: Synchronization via multiplex pulse-train. *IEEE Trans. Circuits Syst. I* 46(9), 1072–1085 (1999)
3. Lee, G., Farhat, N.H.: The bifurcating neuron network 1. *Neural Networks* 14, 115–131 (2001)
4. Hernandez, E.D.M., Lee, G., Farhat, N.H.: Analog Realization of Arbitrary One-Dimensional Maps. *IEEE Trans. Circuits Syst. I* 50(12), 1538–1547 (2003)
5. Izhikevich, E.M.: Simple Model of Spiking Neurons. *IEEE Trans. Neural Networks* 14(6), 1569–1572 (2003)
6. Hamanaka, H., Torikai, H., Saito, T.: Quantized spiking neuron with A/D conversion functions. *IEEE Trans. Circuits Syst. II* 53(10), 1049–1053 (2006)
7. Torikai, H., Tanaka, A., Saito, T.: Artificial Spiking Neurons and Analog-to-Digital-to-Analog Conversion. *IEICE Trans. Fundamentals* E91-A(6), 1455–1462 (2008)
8. Shimakawa, J., Saito, T.: Cyclic D/A Converters Based on Iterated Function Systems. *IEICE Trans. Fundamentals* E87-A(10), 2811–2814 (2004)
9. Saito, T., Shimakawa, J., Torikai, H.: D/A Converters and Iterated Function Systems. In: *Nonlinear Dynamics*, vol. 44, pp. 37–43. Springer, Heidelberg (2006)

Separable Recursive Training Algorithms with Switching Module

Vijanth S. Asirvadam

Intelligent Signal and Image Cluster, Universiti Teknologi PETRONAS,
31750, Bandar Seri Iskandar, Perak D. Ridzuan, Malaysia
Tel.: +6(05)3687881, Fax: +6(05)3657443
vijanth_sagayan@petronas.com.my

Abstract. A novel hybrid or separable recursive training strategies are derived for the training of feedforward neural networks which incorporates a switching module. This new technique for updating weights combines nonlinear recursive training algorithms for the optimization of nonlinear weights with recursive least square type algorithms for the training of linear weights in one integrated routine. The proposed new variant of hybrid weight update includes switching mechanism based on the condition of input data to the system (correlated or noncorrelated). Simulation results demonstrate the improvement of the new proposed switching mode training scheme.

Keywords: Recursive Prediction Error, Multilayer Layer Perceptron (MLP), FeedForward Network, System Identification.

1 Introduction

The training of feedforward neural networks like the multilayer perceptron (MLP) can be considered as an optimization problem where the objective is to minimize cost function, such as the sum-squared error (SSE), with respect to the network parameter (\mathbf{w}) which is.

$$E(\mathbf{w}) = \sum_{k=1}^{N_v} \varepsilon_k^2 = \sum_{k=1}^{N_v} (y_k - d_k)^2 \quad (1)$$

where N_v is the number of training data and y_k and d_k are the actual network output and the desired output respectively for k^{th} training data. The instantaneous error, ε_k , is the difference or error between y_k and d_k .

The MLP-network, $g(\mathbf{w}_r, \mathbf{u}_r)$, is formed by a hidden layer of sigmoidal units and a layer of linear output unit [5], which can be described as

$$y = g(\mathbf{w}_r, \mathbf{u}_r) = \sum_{j=1}^{N_h} \mathbf{w}_j^L f \left(\sum_{i=1}^{N_i} \mathbf{w}_{ij}^{NL} u_i + b_j \right) + d \quad (2)$$

$$f(\gamma_j) = \frac{1}{1 + \exp(-\gamma_j)} \text{ and } \gamma_j = \sum_{i=1}^{N_i} w_{ij}^{NL} u_i + b_j \quad (3)$$

where N_i and N_h are the number of inputs and hidden layer neurons respectively, u_i is the i^{th} element of the input vector \mathbf{u} .

The training strategy employed depends on whether the cost function is minimized with respect to all weights using full nonlinear optimization or by using separable approach in which the nonlinear optimization is applied to weight which are nonlinear-in-weights and the linear training is adopted to the single output neuron which are linear-in-parameter. In offline-training, better minimization can be obtained if the problem is separated with respect to weight orientation (linear/nonlinear) in the network [3][7][10]. Research work in the literatures, including by the present author, on recursive learning techniques for MLP-network showed separable scheme for online neural network (MLP) training significantly outperform the non-separable approaches [1][8]. Recently a new training procedure scheme known as *Extreme Learning Machine* [6], which is subset of separable learning, simplifies the learning rules to updating (only) the weights that are linear-in-parameter by keeping weights connected to nonlinear neuron constant.

This paper proposed an enhancement of hybrid or separable learning technique using a switching module based on the orientation of the input data, correlated or random form. The motivation came from the work by Tae-Hoon *et al.* [11] which proposes a two way training techniques for a offline system using MLP network. This motivates using similar technique on a hybrid MLP network for recursive (or online) weights update.

The paper is organized as follows, section 2 explains on batch algorithms for offline training which form the basis for recursive weight update. Section 3 outlines some recursive trainings while section 4 describes hybrid recursive trainings which includes the variants proposed in this paper with switching module. Finally section 5 presents simulation results using two benchmark problems to demonstrate the improvement gained with the new proposed separable recursive training schemes.

2 Batch Training

The measurement of fit between d_k and y_k can be obtained by the minimization of sum squared of the prediction error, given in equation (1). Assume if batch of data, $\{y_k, d_k\}_{k=1}^N$ is available then the near optimal weight \mathbf{w}^* vector may then be obtained by minimizing the prediction error[4]. This is usually attained by iterative weight vector update procedure given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \lambda_t \mathbf{s}(\mathbf{w}_t) \quad (4)$$

Where λ_t and $\mathbf{s}(\mathbf{w}_t)$ are the learning rate and search direction for the t^{th} iteration. Using backpropagation algorithm[9] which sets the search direction as the negative gradient of the cost function, $E(\mathbf{w})$.

$$\mathbf{s}(\mathbf{w}_t) = -\nabla E(\mathbf{w}_t) \text{ where } \nabla E(\mathbf{w}_t) = \sum_{k=1}^{N_v} \nabla \varepsilon_k^2(\mathbf{w}_t) = \sum_{k=1}^{N_v} \frac{\partial \varepsilon_k^2}{\partial \mathbf{w}_t} \quad (5)$$

Equation (5) can also be written as

$$\nabla E(\mathbf{w}_t) = \sum_{k=1}^{N_v} \nabla y_k(\mathbf{w}_t) \cdot \varepsilon_k = \sum_{k=1}^{N_v} \frac{\partial y_k}{\partial \mathbf{w}_t} \cdot \varepsilon_k \quad (6)$$

The steepest decent normally yields poor convergence rate. In order to improve the minimization of the cost function, a second order properties in form of matrix, $H(\mathbf{w}_t)$, added to the gradient direction to modify the search direction, $\mathbf{s}(\mathbf{w}_t)$. One efficient approach is to approximate the inverse of the hessian matrix, $H(\mathbf{w}_t)$, which is known as Gauss-Newton method

$$M(\mathbf{w}_t) = H^{-1}(\mathbf{w}_t) \text{ where } H(\mathbf{w}_t) = \sum_{k=1}^{N_v} \nabla y_k(\mathbf{w}_t) \nabla y_k^T(\mathbf{w}_t) \quad (7)$$

3 Recursive Training

The recursive training of MLP-network is based on minimization of cost function, $E(\mathbf{w})$, by accumulating information about the distribution given by successive presentation of the training data which are chosen on-line from the training set. Using the one data point available at each t^{th} sample instant, an instantaneous estimate of $E(\mathbf{w})$ is derived as

$$\varepsilon^2(\mathbf{w}_t) = (y(\mathbf{w}_t) - d_t)^2 \quad (8)$$

The stochastic backpropagation (SBP) algorithm performs weight update, \mathbf{w} , at each iteration during the training compared to batch backpropagation which does after running through the entire set of training data. Hence the weight update is given as

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \lambda_t \nabla \varepsilon^2(\mathbf{w}_t) \quad (9)$$

Where λ_t is the learning rate which can be a constant or time varying variable. The random nature of the input data introduces noise during the training of SBP and one possible remedy is to introduce momentum term (the effect of past weight correction).

One disadvantage of SBP is that it shows poor convergence similar to it's batch counterpart and the other main drawback is the tendency of model parameters (especially the bias weights) to continually adapt so that the model output, y , tracks the desired output, d , instead of converging to the desired model weight, \mathbf{w}^* .

The second order algorithms for the recursive approximation of prediction error method are well investigated for network with linear-in-weight.. The same idea can be utilized for identifying time varying nonlinear system by linearizing the network output y_t about the weight \mathbf{w}_t , using equation (2)

$$\nabla y(\mathbf{w}_t) = \frac{\partial}{\partial \mathbf{w}_t} g(\mathbf{w}_t, \mathbf{u}_t) \quad (10)$$

The Hessian matrix for t^{th} training vector can be derived in recursive form as

$$R_t = \alpha_t R_{t-1} + (1 - \alpha_t) (\nabla y(\mathbf{w}_t) \nabla y^T(\mathbf{w}_t)) \quad (11)$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + R_{t-1}^{-1} \nabla y(\mathbf{w}_t) \varepsilon(\mathbf{w}_t) \quad (12)$$

where α_t is the forgetting factor which is usually set to the value between $0.9 \leq \alpha_t \leq 1$. Equation (11)-(12) can thus be considered as recursive approximation of the Gauss Newton search direction or usually known as recursive prediction error (RPE) [4]. In real time implementation the inverse of R is computationally expensive $O(N_w^3)$ to compute and in practice the inverse of the matrix is computed directly as

$$P_t = \frac{1}{\alpha_t} [P_{t-1} - P_{t-1} \nabla y(\mathbf{w}_t) S^{-1}(\mathbf{w}_t) \nabla y^T(\mathbf{w}_t) P_{t-1}] \quad (13)$$

$$\text{Where } S(\mathbf{w}_t) = \alpha_t + \nabla y^T(\mathbf{w}_t) P_{t-1} \nabla y(\mathbf{w}_t) \quad (14)$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + P_t \nabla y(\mathbf{w}_t) \varepsilon(\mathbf{w}_t) \quad (15)$$

4 Hybrid Training

The hybrid recursive training is implemented by separating the training into two categories based on the nature of the neuron (nonlinear/linear). One is to optimize the weights which are nonlinear-in parameter using recursive nonlinear training, described in the earlier section. The second is to implement recursive linear optimization on the output neuron weights which are linear-in-parameter. From equation (6), the decomposed output gradient, $\nabla y(\mathbf{w})$, about the \mathbf{w} can be derived as

$$\nabla y(\mathbf{w}) = \begin{bmatrix} \nabla y(\mathbf{w}^{NL}) \\ \nabla y(\mathbf{w}^L) \end{bmatrix} = \begin{bmatrix} \nabla y(\mathbf{w}^{NL}) \\ \mathbf{r} \end{bmatrix} \quad \mathbf{r} = [r_1 \dots r_{N_h} d] \quad (16)$$

where \mathbf{r} is the vector of hidden layer outputs and the bias of the hidden neuron. Thus the decomposed covariance matrix of estimate \mathbf{w} , P , is given as follows

$$P_{N_w \times N_w} \equiv \begin{bmatrix} P_{N_{NL} \times N_{NL}}^{NL} & 0 \\ 0 & P_{N_L \times N_L}^L \end{bmatrix} \quad \begin{aligned} N_{NL} &= N_i + 1 \cdot N_h \\ N_L &= N_h + 1 \end{aligned} \quad (17)$$

By decomposition of gradient and covariance matrix based on neural network layer which separate the learning to linear/nonlinear parameters three (3) different types (variants) of hybrid training techniques can be derived.

4.1 Type I

The first type (Type I) which is proposed by the present author [1] and Ngia *et al.* [8] does the weight update simultaneously, both weights which are linear and nonlinear-in parameter. The hybrid recursive prediction error (HRPE) training algorithm is composed of RPE and recursive least square (RLS) algorithm for training of nonlinear and linear weights respectively. The training of nonlinear weights using RPE can be summarized as in equation (13)-(15) using the weight vector \mathbf{w}_t^{NL} . The linear weights adaptation using RLS can then be described as

$$S(\mathbf{w}_t^L) = \alpha_t^L + \mathbf{r}_t^T P_{t-1}^L \mathbf{r}_t \quad (18)$$

$$P_t^L = \frac{1}{\alpha_t^L} [P_{t-1}^L - P_{t-1}^L \mathbf{r}_t \mathbf{r}_t^T S^{-1}(\mathbf{w}_t^L) \mathbf{r}_t^T P_{t-1}^L] \quad (19)$$

$$\mathbf{w}_{t+1}^L = \mathbf{w}_t^L + P_t^L \mathbf{r}_t \varepsilon(\mathbf{w}_t) \quad (20)$$

4.2 Type II

The term *Extreme Learning Machine* (ELM) coined by Guang[6] is actually a subset of hybrid form training techniques for feedforward network where weights of the hidden layer of the neural network are set constant thus the hidden layer of the feedforward network act as a nonlinear transformation for the linear output weights to do the global optimization finally. In the ELM technique implementation only the linear weights are updated, which involves equation (18)-(20).

4.3 Type III

A new proposed technique involves the hybrid recursive training which switches according to the correlation of the input data. The switching based training involves decision making module which is,

$$\zeta = \cos^{-1} \left(\frac{\mathbf{u}_{k-1}^T \mathbf{u}_k}{\|\mathbf{u}_{k-1}\| \|\mathbf{u}_k\|} \right) < \varepsilon \quad (21)$$

where it will test the correlation of the input data over each sample instant. The recursive hybrid weight update method proposed in this work will switch between linear and nonlinear update based on the orientation of the input data by using the correlation measure in equation (21). There are two variants of switching module implemented in this paper which are described as follows:

1. Hybrid RPE (HRPE) with switching module I (HRPE-sw(i))

```

if  $\zeta < \varepsilon$  then
    update only the weights linear-in-parameter
else
    update the nonlinear neuron associated
    weights
end

```

2. HRPE switching module II (HRPE-sw(ii))

```

if  $\zeta < \epsilon$  then
    update only the linear-in-parameters weights
else
    update both linear AND nonlinear neuron
    associated weights
end
    
```

Thus the second variant of the proposed switching module always updates the output neuron neuron weights (weights which are linear-in-parameter).

5 Simulation Results

5.1 Test Case I

The following dynamic time series test problem [10] will be used in the evaluation of the hybrid recursive training algorithms. The test case consist of a (3,10,1) MLP-network which is trained to represent a non-linear dynamic system governed by the equation:

$$y_k = 0.3y_{k-1} + 0.6y_{k-2} + 0.6 \sin(\pi u_k) + 0.3 \sin(3\pi u_k) + 0.1 \sin(5\pi u_k) \quad (22)$$

The network input vector u_k is given by:

$$u_k = [y_{k-1}, y_{k-2}, x_{k-1}] \quad (23)$$

The identification process is run over 10,000 iterations with the plant input defined as

$$x_k = \sin\left(2\pi \frac{k}{250}\right) + \sin\left(2\pi \frac{k}{200}\right) \quad (24)$$

5.2 Test Case II

A two tank system (shown in Fig. 1) used for neural modeling purpose as it contains nonlinear dynamic (contains both correlated and randon data) and hence a good example for testing the proposed training algorithms. The system try to predict the level of the second tanks, output h_2 based on the input flow in the first tank, Q_2 , The training set consists of 400 vectors and the MLP network being trained is a (2,20,1) network.

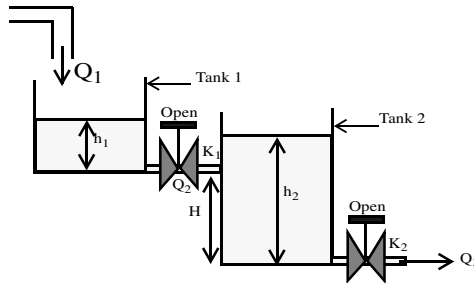


Fig. 1. Two-tank system

5.3 Results Summary and Discussion

The performance measure was evaluated using normalized squared error (NMSE) of batch data by recursively updating the weight vector w_t , given as

$$\sum_k^{N_v} (y_k(w_t) - d_k) \quad / \quad \sum_k^{N_v} d_k^2 \tag{25}$$

Some of the important user defined parameters are initialized where the MLP neural-net weights are set to symmetrical initialization $[w^{NL} \ w^L] = [(0.001+)_{(Ni+1) \times N_h}, (0.001)_{(N_h+1) \times 1}]$, the number of hidden neuron is set to 5, recursive training forgetting factors are set to $\alpha^{NL} = 0.99$, $\alpha^L = 0.99$ and correlation value threshold $\epsilon = 0.05$.

Fig. 2 depict various learning curves obtained by recursively minimizing batch test data, as in equation (32), for dynamic time series- Testcase 1. From the figure plot it is clear that the hybrid training algorithms for MLP neural-net outperform the traditional full nonlinear counterparts (by comparing the RPE). The extreme learning technique (ELM*) did show good convergence at the initialize stage but tend to diverge to infinite value as it reaches the 3000th iteration (sample instant t). Two of the proposed hybrid neural-network training algorithms with switching module perform better than the conventional hybrid technique. Out of the two proposed techniques the second variant (HRPE-sw(ii)) edge slightly better than the first one with faster initial convergence.

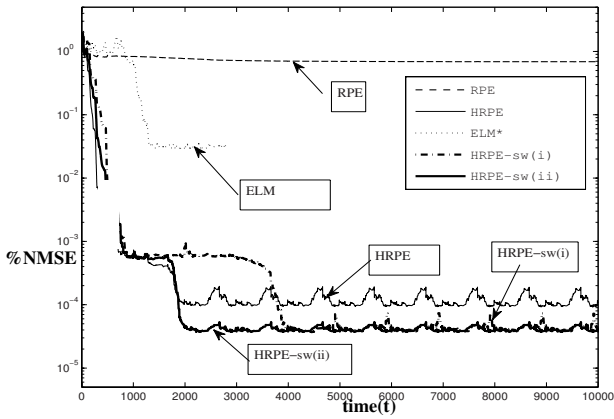


Fig. 2. Learning curves of various hybrid training

Table 1 summarize the performance measure of the various hybrid training algorithms which was discussed in the earlier section of the paper, the RPE methods is included as control measure. The two proposed hybrid methods show the best results in the overall average (mean) and the minimum value for the NMSE measure. The two proposed hybrid methods show the best results in the overall average (mean) and the minimum value for the NMSE.

Table 1. Performance Measure-Learning Curve

Algorithms	%NMSE		
	Mean	Std.-dev	Minimum
RPE	0.6898	0.0022	0.6869
HRPE	1.167e-04	2.554e-05	9.236e-05
ELM	0.0676*	0.1095*	0.0294*
HRPE-sw(i)	4.404e-05	6.241e-06	3.781e-05
HRPE-sw(ii)	4.11e-05	3.229e-06	3.723e-05

ELM*- Stop at 2900th iteration

Fig. 3 and Table 2 summarize the performance of recursive neural-net training algorithms on control benchmark Testcase 2 (using level tank system). The nonlinear system benchmark problem seems difficult to model using RPE and ELM techniques due to its continuous settling and change behaviours. Whereas for the case hybrid training methods a steep convergence in the learning curves being observed. The best results is obtained using the second variant of switching hybrid procedure (HRPE-sw(ii)).

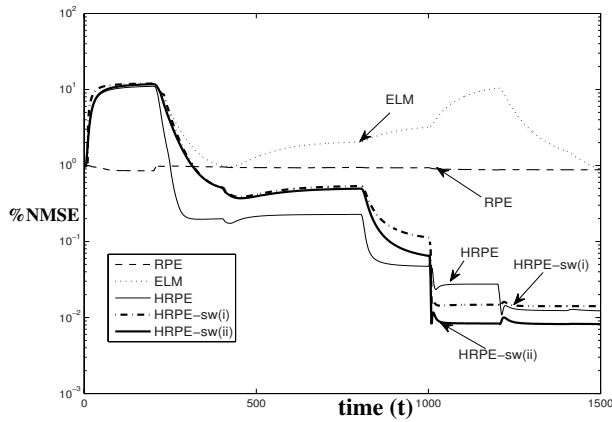


Fig. 3. Learning curves of various hybrid training

Table 2. Performance measure for Testcase II

Algorithms	%NMSE		
	Mean	Std.-dev	Minimum
RPE	0.9006	0.0257	0.8736
HRPE	0.0453	0.0574	0.0108
ELM	4.0513	2.8468	0.8981
HRPE-sw(i)	0.1002	0.1548	0.0141
HRPE-sw(ii)	0.0778	0.1402	0.0082

6 Conclusions

The enhanced form separable recursive least square algorithms for training of feedforward neural network have been proposed in this work. Two new form of separable learning methods with a switching module, tested on MLP-networks show better performance (especially the hybrid second variant) compared to conventional hybrid (or separable) training techniques. The proposed hybrid techniques will only update the weights which are nonlinear-in-parameter if the current input data are uncorrelated to previous input.

Future work will look into the influence of magnitude of prediction error to the linear-in-parameter weights update which can be investigated by looking into the displacement of input vector together with the correlation angle. By limiting the weight correction of recursive linear optimization (RLS) for output neuron weights, a more robust hybrid training method can be obtained, which is in the current stage of investigation by the present author [2] for second order recursive training when the sample arrive at a irregular time interval.

References

- [1] Asirvadam, V.S., McLoone, S.F., Irwin, G.W.: Separable Recursive Training Algorithms for Feedforward Neural Networks. In: IEEE World Congress on Computational Intelligence, Honolulu, Hawaii, May 12-17, pp. 1212–1217 (2002)
- [2] Asirvadam, V.S., Musab, E.J.O.: Wireless System Identification for Linear Networks. In: The 5th International Colloquium on Signal Processing and its Applications CSPA 2009, Kuala Lumpur, Malaysia, March 6-8 (2009)
- [3] Bruls, J., Chou, C.T., Verhaegan, M.: Linear and non-linear system identification using separable least square. In: SYSID 1997, vol. 2, pp. 715–720 (1997)
- [4] Chen, S., Billings, S.A., Grant, P.M.: Nonlinear system identification using neural networks. *Int. Journal of Control* 51(6), 1191–1214 (1990)
- [5] Cybenko, G.: Approximation by superpositions of sigmoidal function. *Mathematics of Signals and Systems* 2, 303–314 (1989)
- [6] Huang, G.B., Zhu, Q.-Y., Siew, C.-K.: Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks. In: 2004 International Joint Conference on Neural Networks (IJCNN 2004), Budapest, Hungary, July 25-29 (2004)
- [7] McLoone, S., Brown, M., Irwin, G., Lightbody, G.: A Hybrid linear/nonlinear training algorithm for feedforward neural network. *IEEE Transaction on Neural Networks* 9(4), 669–684 (1998)
- [8] Lester, N.S.H., Sjöberg, J.: Efficient training of neural nets for nonlinear adaptive filtering using a recursive Levenberg-Marquardt algorithm. *IEEE Transactions on Signal Processing* 48(7), 1915–1927 (2000)
- [9] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J.L. (eds.) *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, vol. 1. MIT Press, Cambridge (1986)
- [10] Sjöberg, J., Mats, V.: Separable Non-Linear Least Squares Minimization- Possible Improvements for Neural Net Fitting. In: *Proceeding of IEEE Workshop in Neural Networks for Signal Processing*, Amelia Island Plantation, Florida, September 24-26, pp. 64–72 (1997)
- [11] Kim, T.-H., Li, J., Manry, M.T.: Evaluation and improvement of two training algorithms. In: *Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, November 3-6, vol. 2, pp. 1019–1023 (2002)

Application-Driven Parameter Tuning Methodology for Dynamic Neural Field Equations

Lucian Alecu¹ and Hervé Frezza-Buet²

¹ CORTEX, LORIA - INRIA Nancy, BP 70239, 54506 Vandoeuvre-lès-Nancy, France

² IMS, SUPELEC, 2, rue Edouard Belin, 57070 Metz, France

{lucian.alecu, herve.frezza-buet}@supelec.fr

Abstract. In this paper, a method is introduced in order to qualify the performance of dynamic neural fields (DNF). The method is applied to Amari's DNF equations, in order to drive the tuning of its free parameters. An original evaluation procedure is presented, and then applied to some input evolution scenarios. Such scenarios define an applicative context, for which the parameters with the lowest evaluation are optimal.

Keywords: dynamic neural field, parameters tuning.

1 Introduction

In the brain of primates, the cortex is a wide neural structure that deals with different kind of information, as vision, audition or motor planning, while it is a quite homogeneous tissue from an anatomical point of view [1]. What may appear as a contradiction is indeed the result of some extremely flexible self-organizing property of the cortical information processing [2,3], allowing computational elements in this neural substrate to be dynamically recruited to cope with everyday behavioral needs, when new skills are to be learnt, or when the body is severely damaged. Understanding this property would allow to design autonomous systems from generic architecture, thus avoiding to face the problem of designing a dedicated module for every single skill needed for the agent's behavior.

In the field of computer science, cortically-inspired self-organization has been addressed by Kohonen, and has lead since then to the famous Self-Organizing-Maps (SOM) [4] vector quantization technique. This exhibits the central role of soft competition in the cortical processing, that can be summarized as follows. At each place of the cortex, an input is provided, that is analysed by some adaptive band-pass filter. The distribution, across the cortical surface, of such filter responses is a rich information, that has to be reduced in order to retain the locally best matching filters. This reduction is a soft competition mechanism, analysed further in the paper. Filters in the winning places adapt slightly in order to increase their matching to current input, and Kohonen has shown that a topologically organized vector quantization emerges from such a learning system.

The soft competition is reported to be the result of some lateral connections in the cortical substrate, made of short range excitatory synapses and longer range inhibitory ones. It emerges from the dynamics of such a neural population of filters, from both their response to the input they receive and the influence of their neighbors. In an attempt to formalize the interaction between the cortical neuronal activities, multiple theoretical models have been proposed, generically being called as *dynamic neural fields* (DNF). The DNF theory has been founded as a specific research field [5]. While many new proposals have been released since then [6,7,8], the Amari model is still regarded as a reference model, and consequently has been successfully used in numerous applications. Nevertheless, the computational power of neural fields has mainly been used for stimuli selection [9] and only few attempts have been made to exploit their properties for self-organization [10], since SOM rather use a computational short-cut for this point. The reason is that parameter tuning for such fields is crucial, and not easy to achieve in such non-linear dynamical systems. This problem motivates the work presented in this paper. Since it is commonly used, our approach is illustrated here, without loss of generality, on the Amari neural field formalism, that is briefly described now. At every time instance t , the evolution of the membrane potential, $u(x, t)$ for each neural unit x of the population (or field) X is expressed by equation (1) introduced in [5]:

$$\tau \frac{du(x, t)}{dt} = -u(x, t) + \int_{x'} w(|x - x'|) f(u(x', t)) dx' + i(x, t) + h \quad (1)$$

where f is a non-linear function (usually a sigmoid), τ and h are real constants, and w is the lateral connections weighting kernel, usually a Mexican-hat function as below:

$$w(r) = A^+ e^{-ar^2} - A^- e^{-br^2} \quad (2)$$

The field conceived in this way reacts to the filter response distribution $i(x, t)$. Typically, the global field response of the Amari model is characterized by the formation of so-called neural “bumps” in some places throughout X , therefore only some patches of neuronal units being highly activated at a given moment in time. In our experiments, the distribution $i(x, t)$ is given a priori, since we rather analyse the field behavior, i.e. the rising of u bumps. In the next section, a quantitative criterion for measuring the field response quality u , in regard to the current i distribution, is introduced. Then, in sections [3] and [4], the measure is applied to Amari fields, and the experimental results are presented, in order to be discussed in section [5].

2 Measuring the Quality of a Field Response

Extending previous work by Mikhailova et al. [11], we formulate here an optimality criterion regarding the dynamics of a neural field, described by the following (P) and (Q) properties :

Property P. *Bumps of a stabilized field response emerge in regions where the input stimuli are locally the strongest and their amplitudes do not depend on the amplitudes of the input stimuli.*

Property Q. *The distance between the center positions of any two bumps of a stabilized field response should stay within bounding limits b_{\min} and b_{\max} , with $b_{\min} < b_{\max}$, in order for the bumps to be neither too sparse, nor too dense in the field.*

A field satisfying $(P \wedge Q)$ develops a selective behavior in any input conditions. Even if such criterion may be regarded rather restrictive, it actually has a strong background motivation. In [12], we show how a field ideally satisfying such properties can support the implementation of self-organizing mechanisms. We present in the following paragraphs a measuring instrument to evaluate whether a field response is satisfying the $(P \wedge Q)$ optimality criterion, in order to drive the parameters choice.

The following analysis is performed for any time instance t , that is omitted for the sake of simplicity. Let us note \mathbb{R}^X the set of functions from X to \mathbb{R} . Both $u(x)$ and $i(x)$ belong to \mathbb{R}^X . The Euclidian distance d on this set is given by equation [3]

$$a \in \mathbb{R}^X, b \in \mathbb{R}^X, d(a, b) = \sqrt{\int_{x \in X} (a(x) - b(x))^2} \tag{3}$$

The (Q) property states that the distance between any two bumps of the field has to lay in the interval (b_{\min}, b_{\max}) . Thereby, let us define $\mathcal{B} \subset \mathbb{R}^X$ as the set of all possible field distributions obtained by placing bumps throughout the field as to satisfy the (Q) condition. In particular, if b_{\min} is very large ($b_{\min} \rightarrow \infty$), \mathcal{B} is the set of distributions formed by a single bump placed at position x , for whichever $x \in X$.

The set of distributions u that satisfy both properties $(P \wedge Q)$ is obviously a subset of \mathcal{B} . Therefore, the functional distance from u to \mathcal{B} (i.e. $d(u, \mathcal{B}) = \min_{u' \in \mathcal{B}} d(u, u')$) should be ideally zero, or practically as small as possible. On the other hand, the (P) property states that $u(x)$ should be high when $i(x)$ is locally the strongest, thus the two distributions should be correlated. This implies that the distance $d(u, i)$ should also be as small as possible. It may also be impossible to have $d(i, u) = 0$, since u should belong to \mathcal{B} and i may not, thus $d(i, u)$ should be reduced to $d(i, \mathcal{B})$. Unless i itself is an element of \mathcal{B} (i.e. $d(i, \mathcal{B}) = 0$), fulfilling the two conditions $(P \wedge Q)$ implies satisfying two opposite constraints, i.e. minimize two different interrelated distances at the same time.

Let us define $\Delta_i^{\mathcal{B}}(u)$ as the residual error of u minimizing the two above distances, a measure of performance of u in regard to i and \mathcal{B} :

$$\Delta_i^{\mathcal{B}}(u) = \sqrt{(d(i, u) - d(i, \mathcal{B}))^2 + d^2(u, \mathcal{B})} \text{ and } \bar{\Delta}_i^{\mathcal{B}}(u) = \Delta_i^{\mathcal{B}}(u)/d(0, \mathcal{B}) \tag{4}$$

To illustrate the intuition behind the introduction of $\Delta_i^{\mathcal{B}}(u)$, let us outline here a geometrical interpretation of this performance measuring instrument. In figure [1], we represent \mathbb{R}^X as the bi-dimensional Euclidian plane. In this plane,

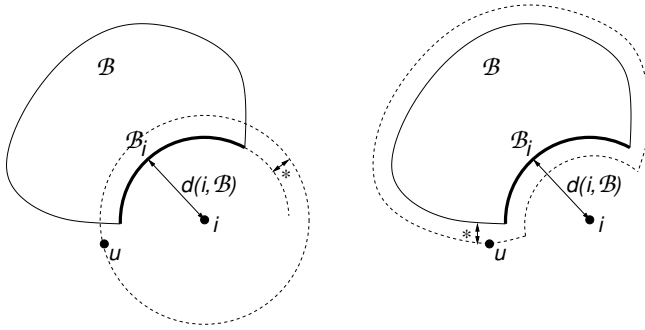


Fig. 1. Geometrical interpretation of $\Delta_i^{\mathcal{B}}(u)$. The thick line represents the set $\mathcal{B}_i = \{i' \in \mathcal{B}, d(i, i') = \min_{u' \in \mathcal{B}} d(i, u')\}$ of field responses satisfying $(P \wedge Q)$ in respect to i , and it indicates a quality reference. u is nearly as qualitative if both following distances marked with the symbol $*$ in the figure are small: $(d(i, u) - d(i, \mathcal{B}))$ (see the left side of the figure) and $d(u, \mathcal{B})$ (see the right side of the figure). $(P \wedge Q)$ is the intersection of the two dotted lined regions in the figure. The proximity of u to \mathcal{B}_i increases while this intersection shrinks around \mathcal{B}_i , i.e. while $\Delta_i^{\mathcal{B}}(u)$ reduces towards zero.

the points stand for a particular distribution of activity over the field, and the Euclidian distance between two points represents the actual distance d defined in equation 3.

In order to implement this theoretical method into a measuring instrument, one has to define the procedure of computing the distance from a distribution (i.e. u or i) to a set of distributions (i.e. \mathcal{B}). Here, this is done by a parametrization of the \mathcal{B} set and then a search through a stochastic gradient descent algorithm. This procedure, that combines a combinatorial exploration for visiting the initial states given to the gradient descent, is not detailed here.

3 Case Study of the Amari Equation

The previously introduced $\bar{\Delta}_i^{\mathcal{B}}(u)$ value allows to drive the parameter settings of some neural fields, taking into account pragmatically the applicative context where soft competition properties are expected. Let us first keep only three degrees of freedom $\alpha_1, \alpha_2, \alpha_3$ from equations 1 and 2, by setting $\tau = 0.3, h = 0, f(x) = (F(x) - F(0))/(F(1) - F(0)), F(x) = 1/(1 + e^{-\alpha_1(2x-1)}), a = 0.35, b = 0, A^+ = \alpha_2, A^- = \alpha_3$, where $\alpha_1 \in \{1, 3, 5, 7, 9\}, \alpha_2 \in [0.2, 1.2]$ (10 steps), $\alpha_3 \in [0, 0.3]$ (30 steps). A grid search exploration of the parameters configurations is made from the previous discrete values. The second element in our experiment is a scenario, that is a succession of i distributions over the field. As we are not in an on-line use of the field, but rather in some experimental and controlled framework, i distribution has to reflect the ones the field would actually have to cope with. This makes the parameter study presented in this section *pragmatic*, i.e driven by the perspective of the actual usage of the soft competition provided by the field in some applicative context. Here, let us consider that we want the

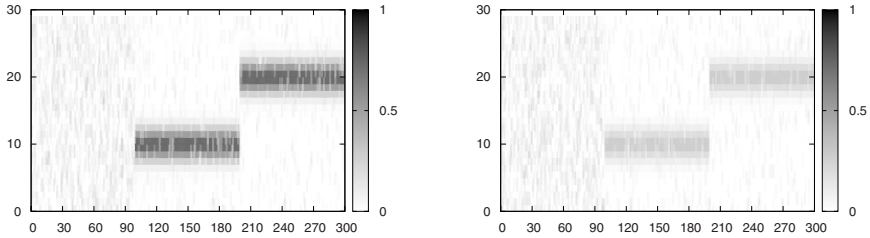


Fig. 2. Scenarios for attention and self-organization. Horizontal axis defines the succession of the time steps. The 1D field X is represented vertically. Each vertical slice represent the input distribution i intensity using a gray scale. In the first 100 time steps, i is a random noise. Then, in the next 100 steps, i has the shape of a noisy Gaussian distribution, whose center changes in the last 100 steps. The two scenarios differ only by the height of the Gaussians.

field to be able to raise a bump in noisy conditions (i.e. start competition from scratch), but also able to reconsider the bump position if some patch of activity exists in the i distribution (i.e. track groups of locally best matching filters). The second requirement is suitable for attentional mechanisms, and both concern self-organization, since competition has to be started from scratch at the beginning of the learning process, where no localized group of filters is dedicated to the current input. The filters have random heterogeneous sensitivities in such initial conditions, a fact that translates into a low and noisy input of the neural field (we do not enter here into the details). This pragmatic context is translated into scenarios in figure 2.

The \mathcal{B} set used to define well formed neural field responses is set to single-bumped fields, i.e. $b_{\min} = \infty$ in the (Q) property. The equation of the bump shape β is $\beta(x, r) = (e^{4(1-|x-r|/r)} - 1)/(e^{4(1-|x-r|/r)} + 1)$, where $x \in [0, 2r]$ and r is the radius of the bump (in our simulations $r = 4$). Allowing several bumps would correspond to have several places able to learn in parallel in the field, which is a challenge for future work, and thus not addressed here.

4 Experiments

Let us first, and mainly, consider here experiments based on the scenario plotted on the left of figure 2. Let us consider all the combination of $\alpha_1, \alpha_2, \alpha_3$ values, when each of them takes the values specified in the previous section (i.e grid search). The performance $\Delta(\alpha_1, \alpha_2, \alpha_3)$ of a field with those parameters is computed as the average, taken over the 300 steps of the scenario, of the $\bar{\Delta}_i^{\mathcal{B}}(u)$ value of the u response of the field. A response that satisfies (P \wedge Q) during the whole scenario would have $\Delta(\alpha_1, \alpha_2, \alpha_3) = 0$, and it would reach $\Delta(\alpha_1, \alpha_2, \alpha_3) = 1$ if the field performs as bad as the null field ($u = 0, \forall x, t$) (see section 2). As the considered scenario consist of three 100-steps parts, note that performing as bad

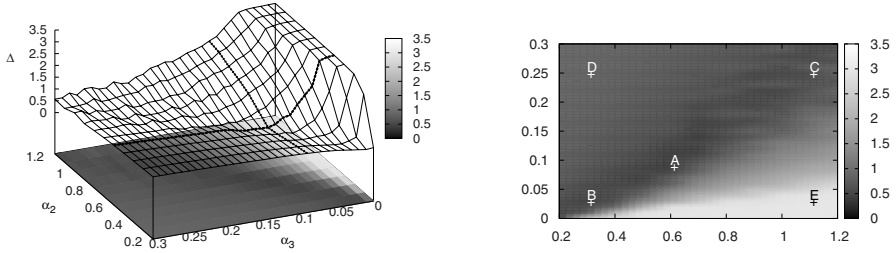


Fig. 3. Left, the value of the quality indicator $\Delta(1, \alpha_2, \alpha_3)$ for all discrete $\alpha_2 \in [0.2, 1.2], \alpha_3 \in [0, 0.3]$ recorded while running the 1D scenario shown in figure 2-left. The best parameters α_2^*, α_3^* are the ones at the crossing of the two thick lines. Right, the specific parameters considered in figure 4. A is the optimum.

as the null response on one of these parts would penalize $\Delta(\alpha_1, \alpha_2, \alpha_3)$ by 0.33. Simulation is made by discretizing equation 1 in time and space, and evaluating each position in a random order, i.e using an *asynchronous update*.

The global minimum is reached for $\alpha_1^* = 1, \alpha_2^* = 0.6, \alpha_3^* = 0.09$. However, if the same evaluation procedure is done for a fixed value of α_1 that is greater than 1, a minimum is obtained for $\alpha_2 = 0.3, \alpha_3 = 0.03$ each time. Therefore, non-linearity has a slight influence on the value of this optimum, since $\Delta(1, \alpha_2^*, \alpha_3^*) = 0.40$ and $\Delta(\alpha_1, 0.3, 0.03) \approx 0.49$ for all other values of α_1 . This means that reducing the non-linearity of the synapses in the Amari model helps for appropriate bump formation. Figure 3-left shows the influence of α_2, α_3 when $\alpha_1 = 1$, and figure 4-upper-left shows the response of the field with optimal $\alpha_1^*, \alpha_2^*, \alpha_3^*$ parameters.

Let us consider non optimal fields on the same scenario, to illustrate the need for accurate parameter settings, for $\alpha_1 = 1$. Figure 4 shows the behavior of the field for different parameter settings. The upper-right frame shows a field that just copies the input. The lower-left one shows a field that is able to raise a bump from noise (i.e. able to trigger learning from scratch), but that is incapable to reconsider the bump position when input changes. The lower-right frame shows a field that can track bumps of input, but that is not able to raise a bump when i is made of a uniform noise. Therefore points B and D from figure 3-right show a similar behavior. As all the $\Delta(\alpha_1, \alpha_2, \alpha_3)$ values are quite close to the optimal ones, since they stand in the flat region of figure 3-left, this shows the difficulty of parameter settings, that needs an accurate search of the right parameter values.

Figure 5-left shows that the quality measurements is robust to different instantiations of the problem (they differ since a random noise is used, as well as a random asynchronous update). This shows that the dramatic input change produces an increase of $\Delta_i^B(u)$, since the field is still in the state fitting previous input, but that it can recover after few steps, due to the change of the u response. The time for recovery could also be measured and then optimized if some application needs highly reactive fields. This point is not investigated here.

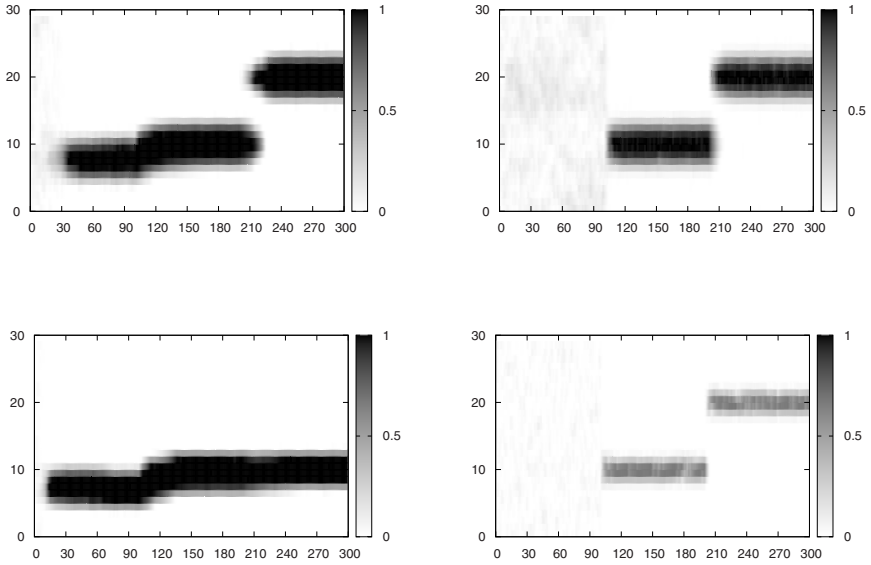


Fig. 4. From upper-left to lower-right, u , the evolution of the field's response to the first scenario, for $\alpha_1 = 1$ and (α_2, α_3) corresponding to the A,B,C and D points from figure 3-right. For point E, the field is not inhibited enough and $u(x, t) = 1, \forall x, t$ (not shown).

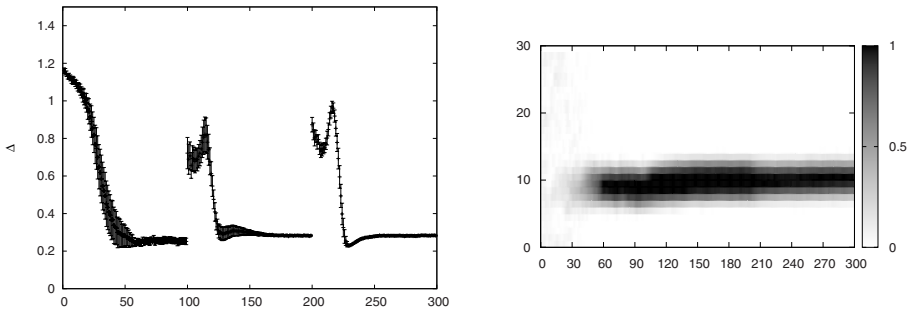


Fig. 5. Left, the scenario is re-run 50 times for the optimal field, allowing to plot at each time step t the mean and standard deviation of $\Delta_i^B(u_t)$. Right, the behavior of the optimal field for the other scenario (plotted in figure 2-right).

5 Discussion

In this paper, an original method that measures the quality of a neural field has been presented. It is based on some controlled scenarios. For Amari equation, the

method appears to be robust to the very scenario instance, and thus significant to the general behavior of the field on such a scenario. Experiments have shown why parameter tuning for Amari equation is hard, since figure 3-left has a flat region where behaviors of the field differs and where optimal parameter stand. Moreover, it appears that the linear synapses, for that equation, helps the field to behave in suitable way. One can notice on figure 5-right that the best field on the second scenario (see figure 2-right) fails in tracking weak bumps of input. This makes the Amari equation unsuitable for driving self-organization [12], whereas first scenario supports its use for attention. This has motivated us to propose a new equation [13] whose extensive analyse by the method presented here is at work.

The authors wish to thank the Region Lorraine for its substantial support to this work.

References

1. Jones, E.: Microcolumns in the cerebral cortex. *PNAS* 97(10), 5019–5021 (2000)
2. Elbert, T., Rockstroh, B.: Reorganization of human cerebral cortex: The range of changes following use and injury. *The Neuroscientist* 10(2), 129–141 (2004)
3. Stavrinou, M., Penna, S., Pizzella, V., Torquati, K., Cianflone, F., Franciotti, R., Bezerianos, A., Romani, G., Rossini, P.: Temporal dynamics of plastic changes in human primary somatosensory cortex after finger webbing. *Cerebral Cortex* 17(9), 2134–2142 (2007)
4. Kohonen, T.: *Self-Organization and Associative Memory*. Springer Series in Information Sciences, vol. 8. Springer, Heidelberg (1989)
5. Amari, S.I.: Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics* 27, 77–87 (1977)
6. Taylor, J.G.: Neural networks for consciousness. *Neural Networks* 10(7), 1207–1225 (1997)
7. Pinto, D., Ermentrout, G.: Spatially structured activity in synaptically coupled neuronal networks: I. traveling fronts and pulses. *SIAM J. Appl. Math.* 62, 206–225 (2001)
8. Pinto, D., Ermentrout, G.: Spatially structured activity in synaptically coupled neuronal networks: II. lateral inhibition and standing pulses. *SIAM J. Appl. Math.* 62, 226–243 (2001)
9. Rougier, N.P., Vitay, J.: Emergence of attention within a neural population. *Neural Networks* 5(19), 573–581 (2006)
10. Ménard, O., Frezza-Buet, H.: Model of multi-modal cortical processing: Coherent learning in self-organizing modules. *Neural Networks* 18(5-6), 646–655 (2005)
11. Mikhailova, I., Goerick, C.: Conditions of activity bubble uniqueness in dynamic neural fields. *Biological Cybernetics* 92(2), 82–91 (2005)
12. Alecu, L., Frezza-Buet, H.: Are neural fields suitable for vector quantization? In: *Proc. of The Seventh International Conference on Machine Learning and Applications (ICMLA 2008)*. IEEE, Los Alamitos (2008)
13. Alecu, L., Frezza-Buet, H.: Reconciling neural fields to self-organization. In: *European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning (ESANN)*, April 2009, pp. 571–576 (2009)

Interspike Interval Statistics Obtained from Non-homogeneous Gamma Spike Generator

Kantaro Fujiwara, Kazuyuki Aihara, and Hideyuki Suzuki

Institute of Industrial Science, University of Tokyo,
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan
{kantaro, aihara, hideyuki}@sat.t.u-tokyo.ac.jp

Abstract. The lengths of interspike intervals between two successive spikes in a neural spike train often vary both within and across trials. In order to describe and analyze neuronal firing, statistical models and methods of probability theory and stochastic point process have been widely applied.

In this study, we compare several non-homogeneous gamma processes on the basis of reproducing the wide distribution of the irregularity statistics *in vivo* and show some conditions for reproducibility. We conclude that the changes of firing rates are not sufficient for describing the fluctuations of the statistics.

Keywords: interspike interval, gamma process, irregularity.

1 Introduction

Firing patterns of cortical neurons look very noisy [1, 2]. Therefore, probabilistic models are necessary to describe such patterns [3, 4]. Baker and Lemon showed that the firing patterns recorded from motor areas can be explained using a continuous-time rate-modulated gamma process [5]. The probability density function of gamma process is depicted as

$$p(T) = \frac{\lambda^\kappa T^{\kappa-1} \exp(-\lambda T)}{\Gamma(\kappa)}, \quad (1)$$

where T denotes an interspike interval, λ denotes a mean firing rate, κ denotes a shape parameter, and $\Gamma(\kappa) = \int_0^\infty T^{\kappa-1} \exp(-T) dT$ is the gamma function. When $\kappa = 1$, gamma process corresponds to Poisson process, and spike train looks irregular. When κ is large, gamma process is approximated by a normal distribution, and when $\kappa \rightarrow \infty$, gamma process corresponds to perfectly-regular firing. Thus, κ is a shape parameter related to regularity.

In examining the model plausibility, reproducibility of the characteristics of real spike train variability is a central problem in the study of brain functions. For quantifying the variability of spike trains, the coefficient of variation C_V is a very common measure which has been widely employed by many researchers [1, 6, 7]. C_V is defined as

$$C_V = \frac{1}{\bar{T}} \sqrt{\frac{1}{n-1} \sum_{i=1}^n (T_i - \bar{T})^2}, \quad (2)$$

where T_i represents the i th ISI, and n the number of ISIs. C_V is a dimensionless index which indicates the spiking irregularity and takes a value 1 for infinitely long purely Poisson series of events, in which event intervals are independently exponentially distributed, and a value 0 for a perfectly periodic sequences.

2 Wide Statistical Distribution

2.1 Statistics of the Data in Vivo

C_V depends on κ in the case of constant firing rate. Baker and Lemon assumed κ to be unique to individual neurons and constant over time [5]. The assumption that κ is unique to individual neurons is also supported by other studies [8, 9]. Unique κ makes the rate of C_V constant in the case of homogeneous gamma process.

However, *in vivo*, C_V distributes widely even though they are recorded from same neuron during same experimental condition in several studies [7, 9]. In our former study [9], C_V distributes widely (0.82 – 1.8) for same neuron under same experimental task (see [9] for a detail) as in Fig. 1. The range of C_V , $0.82 \leq C_V \leq 1.8$ is also valid in other experimental analyses [7].

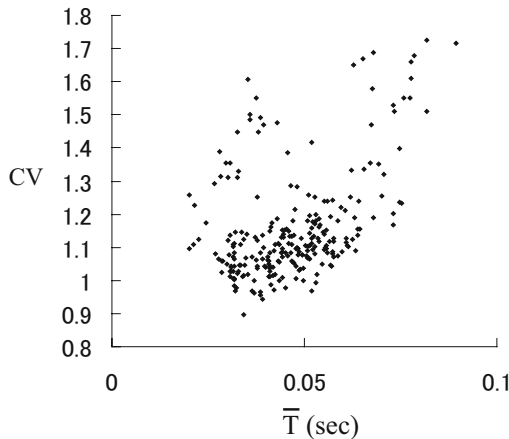


Fig. 1. C_V values obtained from the ISI sequences of the rat gustatory cortex [9]

May this seemingly contradictory phenomenon be explained by varying the mean rate λ which makes the statistics C_V variable? In this paper, we address the question whether the gamma process can reproduce such wide distribution which is often observed in *in vivo* spike data *only* by changing mean firing rates.

2.2 Mechanisms of Wide C_V Distributions

There are three mechanisms for making the distribution of C_V wide *in vivo*: rate fluctuation, irregularity fluctuation, and statistical fluctuation.

Rate fluctuation is a change of the mean firing rates during the experiment, which may make C_V distribution wide. The value of C_V is measured within a time bin with a certain bin size. If the time scale of the rate change is longer than the bin size, C_V can exhibit various values depending on the bins.

Irregularity fluctuation is a change of irregularity factor of a spike generator during the experiment, which may also make C_V distribution wide. It is known that the irregularity measure varies with time [10, 11]. If the irregularity is fluctuating during the experiment, the statistical values can take various values depending on the bins.

Finally, statistical fluctuation may make C_V distribution wide. Due to statistical fluctuation by finite bin size effect, statistical values can take various values.

According to the study of Baker and Lemon [5], fluctuations of the statistics *in vivo* is reproduced by non-homogeneous gamma process which is a gamma process with time-varying firing rate. We confirm whether this is true only by changing λ in equation (1).

3 Numerical Analysis

3.1 Sinusoidally Varying Firing Rate Gamma Process

We first vary the firing rate sinusoidally with period s and its amplitude σ :

$$\lambda(t) = \lambda_0 + \sigma \sin\left(\frac{2\pi t}{s}\right). \quad (3)$$

We set the bin size Δ , and obtained the statistical values C_V from equation (1) and (3). The results for different parameters σ are shown in Fig. 2 and 3.

In Fig. 2, the time scale s of over 3000 millisecond is needed to reproduce the range of C_V from the data *in vivo*. Such long time scale can be seen in delta wave which is a high amplitude brain wave recorded with an EEG and is usually associated with slow-wave sleep [12]. However, delta wave activity during the waking state is not common phenomenon for awake animals [12] and it is impractical to assume the presence of such long time scale dynamics in every experimental data.

If we increase σ from $\sigma = 0.4$ to 0.8 (Fig. 3), the time scale s needed to reproduce the supremum of C_V will be decreased. However, $C_V \leq 1.0$ is irreproducible since large amplitude of rate fluctuation make the rate of C_V large.

From the observation of Fig. 2 and Fig. 3, if we assume the sinusoidally rate modulated gamma process, the dynamics which has fairly long time scale is needed to reproduce the wide C_V distribution, which is an implausible assumption for the real experimental condition.

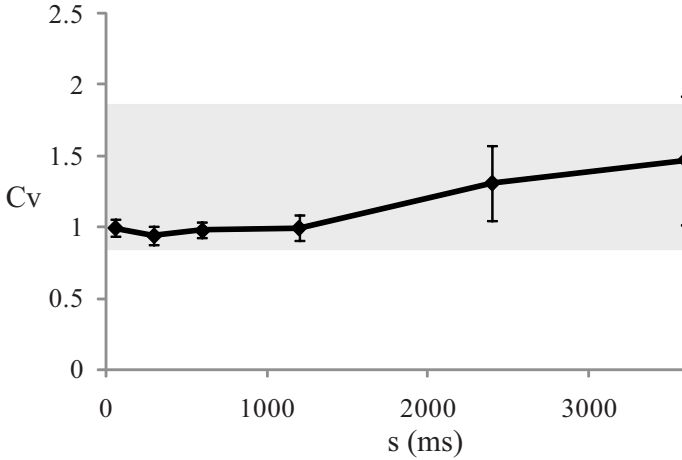


Fig. 2. C_V values obtained from gamma process with sinusoidally varying rate. Hundred C_V rates are obtained from equation (1) and (3), and their means and variances are presented as plots and error bars with different time scale s respectively. The parameters are set as $\kappa = 1, \lambda_0 = 1(\text{Hz}), \Delta = 1000(\text{ms}), \sigma = 0.4$, respectively. Colored area corresponds to the area $0.82 \leq C_V \leq 1.8$ which is observed in vivo (see Fig. [1](#)). If time scale is about $s \sim 3600$, the model can cover the colored area.

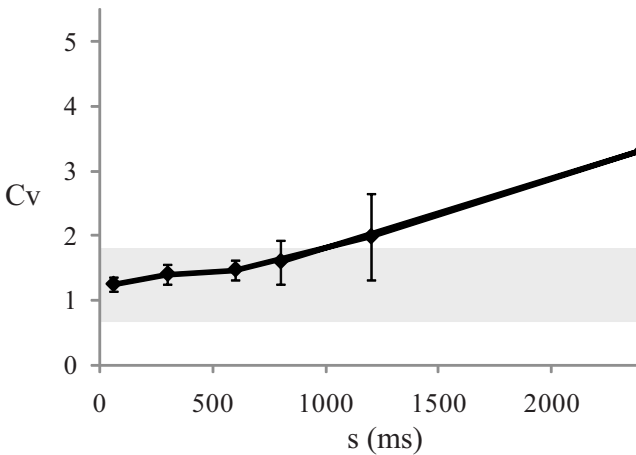


Fig. 3. C_V values obtained from gamma process with sinusoidally varying rate. Hundred C_V rates are obtained from equation (1) and (3), and their means and variances are presented as plots and error bars with different time scale s respectively. The parameter σ is set as $\sigma = 0.8$, and the other parameters are set as the same as in the Fig. 2. Colored area corresponds to the area $0.82 \leq C_V \leq 1.8$ which is observed in vivo (see Fig. [1](#)). If time scale is about $s \sim 800$, the model can cover $C_V = 1.8$, however, $C_V \leq 1.0$ is irreproducible.

3.2 Doubly Stochastic Gamma Process

Next, we consider the doubly stochastic gamma process in which the firing rate is modulated [13]. We consider the case that the random modulation of the firing rate is given by the Ornstein-Uhlenbeck process,

$$\frac{d\lambda}{dt} = -(\lambda - \lambda_0)/s + D\xi(t), \quad (4)$$

where λ is the rate of the gamma process, λ_0 is the mean rate, s is the time scale of the rate change, D is the amplitude of the noise, and $\xi(t)$ is a Gaussian noise with ensemble-averaged quantities $\langle \xi(t) \rangle = 0$ and $\langle \xi(t)\xi(t') \rangle = \delta(t - t')$. The result of the case of doubly stochastic gamma process is shown in Fig. 4.

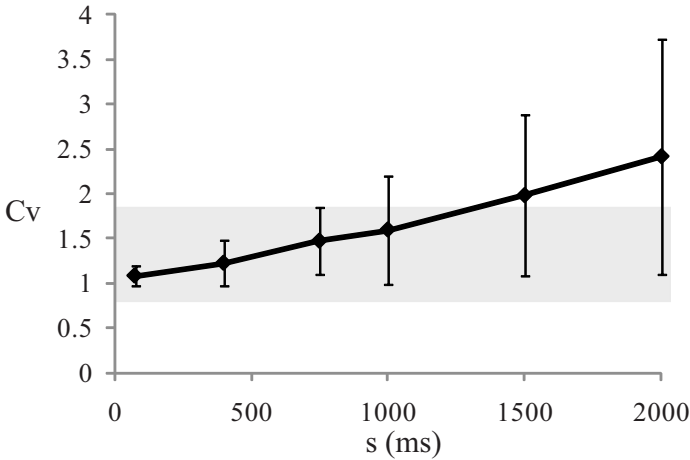


Fig. 4. C_V values obtained from doubly stochastic gamma process. Hundred C_V rates are obtained and their means and variances are presented as plots and error bars with different time scale s respectively. The parameters are set as $\kappa = 1$, $\lambda_0 = 0.02$, $\Delta = 1000$, $D = 0.01$, respectively. Colored area corresponds to the area $0.82 \leq C_V \leq 1.8$ which is observed in vivo (see Fig. 1). If the time scale s is about $s \sim 900$, the model can cover the colored area.

Same as in the case of sinusoidally rate modulated gamma process, the dynamics which has fairly long time scale is needed in doubly stochastic gamma process. According to our observation in this subsection and the previous subsection, an assumption of κ to be unique to individual neurons and constant over time is a *strict* condition in non-homogeneous gamma based model. Therefore, we consider the non-homogeneous gamma process with non-unique κ in the next subsection.

3.3 Non-homogeneous Gamma Process with Varying Shape Factor

We consider the non-homogeneous gamma process with non-unique κ . The firing rate λ and the shape factor of gamma process κ are both modulated sinusoidally with time.

$$\lambda(t) = \lambda_0 + \sigma_\lambda \sin\left(\frac{2\pi t}{s_\lambda}\right), \quad \kappa(t) = \kappa_0 + \sigma_\kappa \sin\left(\frac{2\pi t}{s_\kappa}\right). \quad (5)$$

The result is shown in figure 5.

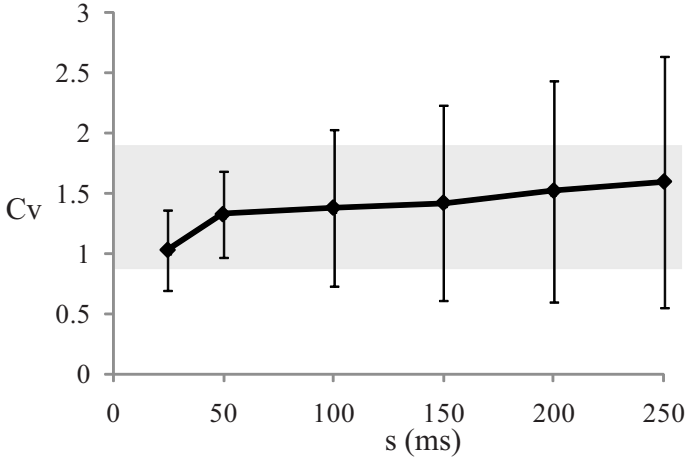


Fig. 5. C_V values obtained from non-homogeneous gamma process with fluctuating κ . Hundred C_V rates are obtained and their means and variances are presented as plots and errorbars with different time scale s_λ respectively. The parameters are set as $\lambda_0 = \kappa_0 = 1$, $\Delta = 1000$, $\sigma_\lambda = \sigma_\kappa = 0.4$, $s_\kappa = 100$, respectively. Colored area corresponds to the area $0.82 \leq C_V \leq 1.8$ which is observed *in vivo* (see Fig. 1). If time scale s is about $s \sim 80$, the model can cover the colored area.

In Fig. 5, time scale s of near 80 millisecond is needed to reproduce the C_V rate from the data *in vivo*. Such time scale can be seen in alpha waves, which is widely observed oscillations in the frequency range of 8 to 12 Hz arising from synchronous and coherent electrical activity [14]. It is plausible to assume the presence of such time scale dynamics in every experimental data. Reproducing the firing statistics of the experimental data is realized by modulating *both* firing rate λ and the shape factor κ of gamma process with a plausible stochastic model.

4 Discussion

We compared several non-homogeneous gamma processes on the basis of reproducing the wide distribution of the irregularity statistics *in vivo*. We conclude

that the assumption of κ to be unique to individual neurons and constant over time is too strict in non-homogeneous gamma based model as in the section 3.1 and 3.2. Instead, we proposed the gamma process based model in which both rate and shape factor modulate with time, and in fact it has broadened the range of the statistics C_V with short time scale.

It has been shown *in vivo* that the changes in the average excitatory synaptic conductance are balanced with those of inhibitory ones in cortical and spinal cord neurons and make irregular firing [15]. Additionally, constant κ is achieved when the ratio of the excitatory and inhibitory activities is constant [16].

According to our result, such ratio may fluctuate with time under the condition of neural balances. Several experimental evidences for such time-changeable balances can be found in recent physiological studies [17, 18]. Time-changeable neural balance enables neuron to assign wide range of statistical values, and may have a possible relationship to robust neural computation in the cortex. It is a future problem to evaluate the effect of such mechanism for improving neural information processing.

Acknowledgments. This research is partially supported by Grant-in-Aid for Scientific Research on Priority Areas 17022012 from MEXT of Japan, and by Grant-in-Aid for JSPS Fellows (20-10814) from Japan Society for the Promotion of Science.

References

- [1] Softky, W.R., Koch, C.: The highly irregular firing of cortical cells is inconsistent with temporal integration of random epsps. *J. Neurosci.* 13, 334–350 (1993)
- [2] Holt, G.R., Softky, W.R., Koch, C., Douglas, R.J.: Comparison of discharge variability in vitro and in vivo in cat visual cortex neurons. *J. Neurophysiol.* 75, 1806–1814 (1996)
- [3] Cox, D.R., Lewis, P.: The statistical analysis of series of events. Methuen, London (1966)
- [4] Tuckwell, H.C.: Introduction to Theoretical Neurobiology, vol. 2. Cambridge University Press, Cambridge (1988)
- [5] Baker, S.N., Lemon, R.N.: Precise spatiotemporal repeating patterns in monkey primary and supplementary motor areas occur at chance levels. *J. Neurophysiol.* 84, 1770–1780 (2000)
- [6] Shadlen, M.N., Newsome, W.T.: The variable discharge of cortical neurons: Implications for connectivity, computation, and information coding. *J. Neurosci.* 18, 3870–3896 (1998)
- [7] Sakai, S., Funahashi, S., Shinomoto, S.: Temporally correlated inputs to leaky integrate-and-fire model can reproduce spiking statistics of cortical neurons. *Neural Networks* 12(8), 1181–1190 (1999)
- [8] Shinomoto, S., Shima, K., Tanji, J.: Differences in spiking patterns among cortical neurons. *Neural comput.* 15, 2823–2842 (2003)
- [9] Fujiwara, K., Fujiwara, H., Tsukada, M., Aihara, K.: Reproducing bursting interspike interval statistics of the gustatory cortex. *Biosystems* 90, 442–448 (2007)

- [10] Davies, R.M., Gerstein, G.L., Baker, S.N.: Measurement of time-dependent changes in the irregularity of neural spiking. *J. Neurophysiol.* 96, 906–918 (2006)
- [11] Fujiwara, K., Aihara, K.: Time-varying irregularities in multiple trial spike data. *European Physical Journal B* 68(2) (2009)
- [12] Maquet, P., Degueldre, C., Delfiore, G., Aerts, J., Péters, J.M., Luxen, A., Franck, G.: Functional neuroanatomy of human slow wave sleep. *The Journal of Neuroscience* 17(8) (1997)
- [13] Cox, D.R., Isham, V.: *Poisson processes*. Chapman & Hall, Boca Raton (1980)
- [14] Semba, K., Komisaruk, B.R.: Neural substrates of two different rhythmical vibrissal movements in the rat. *Neuroscience* 12(3) (1984)
- [15] Shu, Y., Hasenstaub, A., McCormick, D.A.: Turning on and off recurrent balanced cortical activity. *Nature* 423 (2003)
- [16] Miura, K., Tsubo, Y., Okada, M., Fukai, T.: Balanced excitatory and inhibitory inputs to cortical neurons decouple firing irregularity from rate modulations. *The Journal of Neuroscience* 27(50) (2007)
- [17] Dani, V.S., Chang, Q., Maffei, A., Turrigiano, G.G., Jaenisch, R., Nelson, S.B.: Reduced cortical activity due to a shift in the balance between excitation and inhibition in a mouse model of rett syndrome. *Proceedings of the National Academy of Sciences of the United States of America* 102(35) (2005)
- [18] Heiss, J.E., Katz, Y., Ganmor, E., Lampl, I.: Shift in the balance between excitation and inhibition during sensory adaptation of s1 neurons. *The Journal of Neuroscience* 28(49) (2008)

A Novel Method for Progressive Multiple Sequence Alignment Based on Lempel-Ziv

Guoli Ji^{1,*}, Congting Ye¹, Zijiang Yang², and Zhenya Guo¹

¹ Department of Automation, Xiamen University, 361005 Xiamen, China
glji@xmu.edu.cn, yecongting123@yahoo.com.cn, uoxiaoya1@gmail.com

² School of Information Technology, York University, Toronto, Canada M3J 1P3
zyang@mathstat.yorku.ca

Abstract. In this paper, we propose LZ_MSA, a novel method for progressive multiple sequence alignment based on Lempel-Ziv. The vector space is constructed by 10 types of copy modes. Under this approach, sequence alignment is converted into vector alignment and the guide tree can be dynamically amended. Finally we use five subsets in the standard dataset of BALiBASE to validate the proposed algorithm. Compared to ClusatalW, MAFFT, LZ_MSA reduces the alignment time without sacrificing accuracy.

Keywords: Multiple Sequence Alignment, Time Complexity, LZ_MSA.

1 Introduction

The Biological problem of multiple sequence alignment has been proved to be an NP-complete problem [1]. The bottleneck of this method is the large amount of calculation in pairwise sequence alignment. Therefore, how to do effective pairwise sequence alignment is the main issue in this paper.

Currently, there have been many algorithms to solve the multiple sequence alignment problems. One is based on the random search strategy such as hidden Markov model [2], simulated annealing algorithm [3, 4], genetic algorithm [5-8], and etc. These algorithms have been proved to be flexible and effective. However, they are not very stable and the computation time is long. The progressive alignment algorithm has been introduced into multiple sequence alignment problems. The basic idea of these algorithms is the iterative use of the two dynamic programming sequence alignment algorithms. The alignment begins with two sequences, and then new sequence will be gradually added until all sequences are added in. However, adding new sequences in different order will produce different results. Therefore, the identification of suitable alignment order is a key issue. Thompson etc. proposed a ClustalW algorithm, which is the most widely used progressive multiple sequence alignment algorithm [9]. The evolutionary information was introduced into the alignment process - through the construction of the guide tree to make multiple sequence alignment a gradual pairwise alignment process. In this way, the amount of calculation will be reduced and it is very practical. Generally, ClustalW is a successful method for multiple sequence alignment. Nevertheless, it can easily be

* Corresponding author.

trapped into local optimal solution since it greedily follows the guide tree to add all the sequences together. Later, many algorithms have improved the ClustalW algorithm. Gotoh proposed the idea of iteration to improve the multiple sequence alignment accuracy [10]. Kazutaka Katoh proposed MAFFT algorithm which combined the fast Fourier transform and the guide tree [11].

Lempel-Ziv algorithm (LZ) is a dictionary compression algorithm proposed by Lempel and Ziv in 1976 [12]. The complexity of the symbolic sequence they defined refers to the minimum steps to generate the sequence through the adoption of the most copied and additional "adding" a character from the null string [13-14]. The complexity of symbolic sequence reflects the similarity situation of sequences.

This paper introduces LZ_MSA, a novel method for progressive multiple sequence alignment based on Lempel-Ziv. In the pairwise sequence alignment, it converts the sequence alignment into vector alignment, which can prevent the complicated calculation. In this way, the calculation time is reduced. And then the dictionary strategy is applied, which also helps to reduce the calculation time. The dynamically amending of the guide tree can avoid trapping into local optimal solution. It also improves the accuracy of the algorithm. Finally the time complexity of LZ_MSA is $O(N \log N)$ compared to ClustalW's $O(N^2 L^2)$. Therefore, our method effectively improves the speed of pairwise sequences alignment without scarifying the accuracy.

2 The Progressive Multiple Sequence Alignment LZ_MSA

ClustalW is a classical progressive multiple sequence alignment method. It constructs the distance matrix first and then generates the guide tree in accordance with the calculation of the distance matrix. With the guidance of guide tree from the beginning of most closely two sequences, it gradually introduces a new sequence until all the sequences are added in. The method makes the guide tree remain unchanged and the generated sequence can only be compared by the fixed order. However, because of its large amount of calculation in pairwise sequence alignment and the greed alignment strategy, its speed and accuracy is limited. In order to overcome the above-mentioned limitations, a novel progressive multiple sequence alignment LZ_MSA is developed. The process of LZ_MSA is as follows:

Step 1. Convert the DNA sequences into a vector space

Assume the set of aligned sequences are $S = \{s_1, \dots, s_N\}$, where s_i is the i th sequence and $i \in \{1, \dots, N\}$. We convert the sequence to the vector space. Now we introduce the specific vector conversion process: the positive replication of DNA sequences is a kind of identical permutation on the set $M = \{A, G, C, T\}$:

$$(1) p(A) = A, p(T) = T, p(G) = G, p(C) = C ;$$

while the complementary of DNA sequences is another kind of permutation:

$$(2) p(A) = T, p(T) = A, p(G) = C, p(C) = G ;$$

Based on the effective combination of positive replication and complementary for DNA sequences, we can get the other eight kinds of replication as follows:

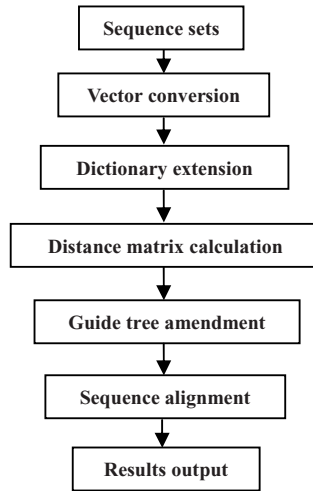


Fig. 1. LZ_MSA flowchart

Each replication is based on the above identical and complementary transformation, the replication only contains complementary transformation also have two kinds:

$$(3) p(A) = G, p(G) = A, p(C) = T, p(T) = C;$$

$$(4) p(A) = C, p(C) = A, p(G) = T, p(T) = G;$$

The following replications are the combination of identical and complementary transformation:

$$(5) p(A) = A, p(G) = G, p(T) = C, p(C) = T;$$

$$(6) p(A) = A, p(C) = C, p(G) = T, p(T) = G;$$

$$(7) p(A) = A, p(T) = T, p(G) = C, p(C) = G;$$

$$(8) p(G) = G, p(T) = T, p(A) = C, p(C) = A;$$

$$(9) p(G) = G, p(C) = C, p(A) = T, p(T) = A;$$

$$(10) p(C) = C, p(T) = T, p(A) = G, p(G) = A;$$

Among the four bases of the six kinds of replications above, there are two for identical transformation, two for complementary transformation.

Thus, any DNA sequence may be achieved from a null sequence. Hence, there are ten time complexities for a particular DNA sequence, which constitutes a vector with ten components, denoted as $(c_1(s), c_2(s), \dots, c_{10}(s))$. Considering the corresponding relationship between DNA sequences and vectors above, the comparison of DNA sequences may be transformed to comparison of vectors. The correlated value of any two given species s_1 and s_2 can be computed by cosine of their corresponding vectors:

$$c(s_1, s_2) = \frac{\sum_{i=1}^{10} c_i(s_1) \times c_i(s_2)}{\left[\sum_{i=1, j=1, j \neq i}^{10} c_i^2(s_1) \times c_j^2(s_2) \right]^{\frac{1}{2}}} \tag{1}$$

Where $s_1 = (c_1(s_1), c_2(s_1), \dots, c_{10}(s_1))$, $s_2 = (c_1(s_2), c_2(s_2), \dots, c_{10}(s_2))$. Then we can get the distance of s_1 and s_2 :

$$d(s_1, s_2) = 1 - c(s_1, s_2) \tag{2}$$

For example, Sequence $s_1 : ACGGTC$, $s_2 : ACGATC$ and $s_3 : GGTGTT$ can be converted into vector space:

$s_1'(5, 4, 4, 4, 5, 5, 5, 5, 5, 5)$, $s_2'(5, 4, 4, 4, 5, 4, 5, 5, 5, 5)$, $s_3'(3, 4, 3, 4, 3, 3, 3, 3, 3, 3)$. From Equation 2, $d(s_1', s_2') = 0$ and $d(s_1', s_3') = 0.03$. It is obvious to observe that the distance between s_1 and s_2 is closer and the distance between s_3 and s_1 is further.

Step 2. Expand the dictionary, compute the distance matrix, construct and amend the guide tree

Following step 1, we can get the vector set $S' = \{s_1', \dots, s_n'\}$. Initially, the dictionary $G_m^1 = \Phi$ is empty and a random fragment s_1' is set to be the first residue of the corresponding sequence. According to Equation 2, $d(s_1, s_2), d(s_1, s_3), \dots, d(s_1, s_n)$ can be calculated as the residue of sequences s_2, s_3, \dots, s_n . These directories are ordered from small to large. Then the maximum value is $d_{\max}(s_1, s_x)$ and the initial dictionary directory is from 0 to $d_{\max}(s_1, s_x)$. Thus, the LZ dictionary is created. To further reduce the execution time, D is only partially calculated as follows: an initial sequence is selected and compared to all the other sequences. The resulting distances are split evenly into two groups based on d, one containing the smallest distances, denoted by $d \leq \frac{d_{\max}(s_1, s_x)}{2}$, and the other containing the largest distances, denoted by $d > \frac{d_{\max}(s_1, s_x)}{2}$. The process is repeated recursively on each group until the number of sequences in a group is two. The benefit is that only $N \log(N)$ distances need to be calculated.

When a new sequence s_x' is added, it is considered to be a new directory element. It is necessary to expand the existing dictionaries, which indicates that the new dictionary is $G_m^{N+1} = G_m^N \cup \{s_x'\}$. Then reset $s_x' = \Phi$. In accordance with this situation, followed by the addition of new sequences, the evolutionary distance matrix will be updated by the newly added sequence.

Distance matrix method is a mature algorithm to develop guide tree. The common methods are: UPGMA [17] □Fitch-Margoliash [18] and NJ (Neighbor-joining Method) [19]. All of these methods can generate root trees, and NJ algorithm is the most efficient algorithm. In this paper, we use the NJ algorithm which is based on the distance method to construct the guide tree. The distance matrix is calculated by step 2. Next we put the distance matrix into the Neighbor program in PHYLIP. Finally the results of the evolutionary relationship - the guide tree will be produced. LZ_MSA will dynamically amend the guide tree and avoid trapping into local optimal solution.

Step 3. Sequence alignment in accordance with the guide tree

Followed by step2, in terms of the selection of the alignment scoring system in the process of sequence alignment, the normalized similar matrix and gap penalty proposed by MAFFT have achieved good results in practical applications [11]. Therefore, this method will be used in this paper. Similarity matrix \hat{M}_{ab} can be represented as:

$$\hat{M}_{ab} = [(M_{ab} - average2)/(average1 - average2)] + S^n$$

Where $average1 = \sum_0^a f_a M_{aa}$, $average2 = \sum_a^b f_a f_b M_{ab}$, M_{ab} is raw similar matrix,

f_a is the frequency of occurrence of amino acids A, and S^a is a parameter that functions as a gap extension penalty. Gap penalty can be represented as

$G(i, x) = S^{op} \cdot \{1 - [g^{start}(x) + g^{end}(i)]/2\}$, Where S^{op} corresponds to gap opening penalty, $g^{start}(x)$ is the number of the gaps that start at the x the site, and $g^{end}(i)$ is the number of the gaps that end at the x the site.

The program of LZ_MSA has been carried out in the Windows system in VC++ language, and the program also integrates ClustalW, MAFFT with LZ_MSA for comparison.

3 Results and Discussions

The proposed algorithm is tested using the standard dataset in Bali BASE. The Bali BASE is a protein multiple sequence alignment sets, which contain 144 test cases with more than 1000 sequences. According to the different characteristics of sequences, the alignment cases were divided into five subsets [19], as shown in Table 1.

The test results are evaluated by Bali score [19]. Bali score is a program to evaluate the merits of the alignment, which belongs to Bali BASE. It evaluates the alignments based on measured values of the SPS (residue of the number). Assume that the sequence number is N, each sequence has M columns, and the column of reference sequence is M_r , the ith residue is expressed as $C_{i1}, C_{i2}, \dots, C_{iN}$. Then the calculation of two values is respectively described as follows:

Table 1. The features of five subsets in Bali BASE

Bali BASE	Each feature of subsets
Ref1	The lengths of the sequences are the same
Ref2	One set includes more than 15 close kin sequences and an orphan sequence
Ref3	The sequences include more than one family and each family is far between genetic sequence of number
Ref4	The sequences have N/C terminal extension feature (reach 400 residues)
Ref5	The sequences have long internal insertion

SPS: For each pair residues c_{ij} and c_{ik} in one column, define P_{ijk} , if c_{ij} can match the same as c_{ik} , we set P_{ijk} to 1, or else to 0. The formula of S_i in each column is as follows:

$$S_i = \sum_{j=1, j \neq k}^N \sum_{k=1}^N P_{ijk} \quad (3)$$

Assuming the value of S_r corresponds to the value of S_i in reference data, so the formula of SPS is as follows:

$$SPS = \sum_{i=1}^M S_i / \sum_{i=1}^{M_r} S_{ri} \quad (4)$$

At the same time, Thompson has also provided the scores of other popular alignment evaluation methods. Bali score and these scores can be downloaded free of charge from the web site:

(<http://bips.ustrasbg.fr/en/Products/Databases/BAlIbASE/progscores.html>)

Table 2. Comparison of the SPS average and the running time for each subset

	ClustalW	MAFFT	LZ_MSA
Ref1	0.871	0.782	0.735
Ref2	0.498	0.766	0.767
Ref3	0.517	0.631	0.608
Ref4	0.672	0.574	0.598
Ref5	0.683	0.633	0.585
Time(s)	2202	1466	530

Table 2 shows the comparison of alignment results and running time. It can be observed that LZ_MSA can test all test cases in the five subsets. Overall, the results with LZ_MSA are better than ClustalW. In general, it has the same effect as MAFFT. For comparison, for Ref1, Ref4 and Ref5, LZ_MSA is slightly worse than ClustalW, which

shows that for the continuous insert circumstances, LZ_MSA may not produce the best results. For Ref2 and Ref3, LZ_MSA is obviously superior to ClustalW, which indicates that for a family protein (or high similarity sequence), no matter whether the alignment is between families or between family and orphans, LZ_MSA can produce superior results. The quality of the alignment results was mainly due to the fact that whether the guide tree is reasonable. From the above, it shows that LZ_MSA reduces the alignment time without sacrificing the accuracy.

4 Conclusion

It can be seen from the above analysis that LZ_MSA reduces the running time of pairwise comparison compared to other methods such as ClustalW and MAFFT. At the same time, it can keep the same accuracy level and under some occasions the accuracy level is higher. More importantly, it converts the sequence alignment into vector alignment, which can prevent the complicated calculation. It can also avoid trapping into local optimal solution by dynamically amending of the guide tree. Finally, combining the LZ algorithm with the multiple sequence alignment method is closer to the biology truth. Moreover, there is still some room for improvement. The possible future work includes how to convert sequence, and how to calculate similar matrix and the gap penalty.

Acknowledgment

The authors thank Dan Zou for his valuable assistance. This project was supported by funds from the National Natural Science Foundation of China (No. 60774033), Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20070384003), and the Key Research Project of Fujian Province of China (No. 2009H0044), all to GJ. This project is also partially supported by Natural Sciences and Engineering Research Council of Canada.

References

1. Russell, D.J., Otu, H.H., Sayood, K.: Grammar-based distance in progressive multiple sequence alignment. In: BMC Bioinformatics (2008)
2. Notredame, C.: Recent Evolutions of Multiple Sequence Alignment Algorithms. *PLoS Computational Biology* 3(8), 1405–1408 (2007)
3. Kim, J., Pramanik, S., Chung, M.J.: Multiple sequence alignment using simulated annealing. *BMC Bioinformatics* 10, 419–426 (1994)
4. Kim, J., Cole, J.R., Pramanik, S.: Alignment of possible secondary structures in multiple RNA sequences using simulated annealing. *BMC Bioinformatics* 12, 259–267 (1996)
5. Anbarasu, L.A., Narayanasamy, P., Sundararajan, V.: Multiple Sequence Alignment using parallel genetic algorithm. In: *The Second Asia Pacific Conference on Simulated Annealing*, Canberra, Australia (1998)
6. Gonzalez, R.R., Izquierdo, C.M., Seijas, J.: Multiple Protein Sequence comparison by genetic algorithms. In: *SP IE298* (1999)

7. Zhang, C., Wong, A.K.C.: A genetic algorithm for multiple molecular sequence alignment. *BMC Bioinformatics* 13, 565–581 (1997)
8. Notredame, C., Higgins, D.G.: SAGA: sequence alignment by genetic algorithm. *Nucleic Acids Research* 24, 1515–1524 (1996)
9. Thompson, J.D., Higgins, D.G., Gibson, T.J.: CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research* 22, 4673–4680 (1994)
10. Gotoh, O.: Significant Improvement in Accuracy of Multiple Protein Sequence Alignments by Iterative Refinement as Assessed by Reference to Structural Alignments. *Journal of Molecular Biology* 264, 823–838 (1996)
11. Katoh, K., Misawa, K., Kuma, K.-i., Miyata, T.: MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Research* 30, 3059–3066 (2002)
12. Lempel, A., Ziv, J.: On the complexity of finite sequences. *IEEE Transactions on information theory* 22(1) (1976)
13. Feng, D.F., Doolittle, R.F.: Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Molecular Biology and Evolution* 25(4), 351–360 (1987)
14. Gotoh, O.: Significant improvement in accuracy of multiple protein sequence alignment by iterative refinement as assessed by reference to structural alignment. *Journal of Molecular Biology* 264, 823–838 (1996)
15. Zheng, X., Li, C., Wang, J.: A complexity-based measure and its application to phylogenetic analysis. *Journal of Mathematical Chemistry* 43, 26–31 (2008)
16. Fitch, W.M., Margoliash, E.: Construction of phylogenetic trees. *Science* 155, 279–284 (1967)
17. Saitou, N., Nei, M.: The neighbor-joining method: a new method for reconstructing phylogenetic tree. *Molecular Biology and Evolution* 4, 406–425 (1987)
18. Carrillo, H., Lipman, D.: The multiple sequence alignment problem in biology. *SIAM Journal on Applied Mathematics* 48(5), 1073–1082 (1988)
19. Thompson, J.D., Plewniak, F., Poch, O.: A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Research* 27(13), 2689–2690 (1999)

Variational Bayes from the Primitive Initial Point for Gaussian Mixture Estimation

Yuta Ishikawa¹, Ichiro Takeuchi¹, and Ryohei Nakano²

¹ Nagoya Institute of Technology, Dept. of Scientific and Engineering Simulation,
Gokiso-cho, Showa-ku, Nagoya, Aichi 466-8555 Japan

ishikawa@goat.ics.nitech.ac.jp, takeuchi.ichiro@nitech.ac.jp

² Chubu University, Dept. of Computer Science,
Mastumoto-cho 1200, Kasugai, Aichi 487-8501 Japan

nakano@cs.chubu.ac.jp

Abstract. Gaussian mixture model (GMM) is one of the important models to approximate probability distributions. There are various methods for Gaussian mixture estimation such as the EM algorithm, sampling method, and the Bayes method. In this paper, we are concerned with the Gaussian mixture estimation problem using the variational Bayes (VB), which is an approximation of the Bayes method. In the VB, it is important to choose its initial values carefully since the objective function of the problem is multimodal. In this paper, we propose a method which employs *primitive initial point (PIP)* as an initial value of the VB and performs multi-directional search from the PIP. We present the motivation and rationale of our method and demonstrate its effectiveness through numerical experiments using real data sets.

Keywords: Gaussian mixture estimation, variational Bayes, primitive initial point, deterministic annealing.

1 Introduction

Gaussian mixture model (GMM) is widely used in many applications because it can approximate various forms of probability distributions [1]. Many approaches have been proposed for GMM estimation problem, such as EM algorithm, sampling method and Bayesian method. Recently, the variational Bayes (VB) method which is an approximation of the Bayes method based on the mean field approximation [2,3,4,5]. In this paper we apply the VB method [6] to GMM estimation problem. In the VB, one can only find a local optimum because the free energy function of the problem is multimodal with respect to the parameters. Therefore, we should choose initial points carefully to find an excellent solution. Deterministic annealing is often useful for finding a better local solution [7]. Recently, Katahira et al [8] adapted deterministic annealing approach to the VB, which is called the DAVB method. They empirically demonstrated that the DAVB has the ability to find an excellent solution.

In this paper we propose an alternative method to challenge the local optimality problem. The starting search point of our algorithm is the same as that of the

DAVB. We define the optimal solution of the DAVB at the highest temperature as *the primitive initial point (PIP)*. We investigate the curvature of the free energy function at the PIP. As we will see later, the Hessian matrix of the original (not annealed) free energy function at the PIP has both positive and negative eigenvalues, showing the PIP is a saddle point. In addition, we obtained empirical evidence [9] that the PIP is an excellent starting point. In particular, we examined how the negative free energy function changes along the straight path from the PIP to excellent solutions, and found that the negative free energy function is monotonically increasing through the path in most (but not all) cases. These results imply that we can obtain excellent solutions without annealing if we start searching from the PIP in the direction of increasing the negative free energy function. Using these empirical results, we develop an efficient multi-directional search strategy for VB-based GMM estimation problem. The computational cost of our approach is comparatively small because it does not need an annealing process. Another advantage of our approach is that the algorithm is deterministic, while other approaches, including the DAVB, have random feature.

This paper is organized as follows. Section 2 describes the variational Bayes (VB) method and the deterministic annealing VB (DAVB) method for Gaussian mixture model. In section 3, we define the PIP and analyze the curvature of the free energy function at the PIP. Using the implication from the analysis, we develop a multi-directional search algorithm from the PIP. In section 4, we evaluate the performance of our method using real data sets. Finally, we conclude the paper in section 5.

2 Background

2.1 Gaussian Mixture Model

Let us formulate Gaussian mixture model (GMM) estimation problem. Consider D -dimensional K -class Gaussian mixture model. Let $\{\mathbf{x}_n | \mathbf{x}_n \in \mathbb{R}^D, n = 1, \dots, N\}$ be an observed data set. In GMM, we assume that \mathbf{x}_n is generated from the mixture of multivariate Gaussian distributions:

$$p(\mathbf{x}_n | \boldsymbol{\theta}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \mathbf{A}_k^{-1}),$$

where π_k , $\boldsymbol{\mu}_k$ and \mathbf{A}_k are the mixing coefficient, the mean vector and the precision matrix of the k^{th} component, respectively. The objective is to estimate model parameters $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\mu}_k, \mathbf{A}_k\}, k = 1, \dots, K$ from the given data set.

2.2 Variational Bayes Method

In GMM estimation, the number of mixture components K is generally unknown. The variational Bayes (VB) method provides a convincing way to estimate the appropriate number of mixture components from data [23]. The VB method is an approximation of the Bayes method using the mean field approximation [45].

In the VB framework, the lower bound of a marginal log-likelihood $\ln p(\mathbf{X})$ is maximized. The lower bound $\mathcal{L}(Q(\mathbf{Z}, \boldsymbol{\theta}))$ is derived as

$$\begin{aligned} \ln p(\mathbf{X}) &= \sum_{\mathbf{Z}} \ln \int p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta}) d\boldsymbol{\theta} = \sum_{\mathbf{Z}} \ln \int Q(\mathbf{Z}, \boldsymbol{\theta}) \frac{p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta})}{Q(\mathbf{Z}, \boldsymbol{\theta})} d\boldsymbol{\theta} \\ &\geq \sum_{\mathbf{Z}} \int Q(\mathbf{Z}, \boldsymbol{\theta}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta})}{Q(\mathbf{Z}, \boldsymbol{\theta})} \right\} d\boldsymbol{\theta} \\ &\equiv \mathcal{L}(Q(\mathbf{Z}, \boldsymbol{\theta})), \end{aligned} \tag{1}$$

where \mathbf{Z} denotes latent variables and $Q(\mathbf{Z}, \boldsymbol{\theta})$ is an arbitrary distribution called the variational posterior distribution. Through the maximization of the lower bound eq.(1), we approximate the true posterior distribution by $Q(\mathbf{Z}, \boldsymbol{\theta})$. To apply the VB method to GMM estimation, the joint distribution $p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta})$ is decomposed as [10]:

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta}) = p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda})p(\mathbf{Z}|\boldsymbol{\pi})p(\boldsymbol{\pi})p(\boldsymbol{\mu}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda}). \tag{2}$$

In general, conjugate priors are employed as prior distributions; the Dirichlet distribution for $p(\boldsymbol{\pi})$ and the Gaussian-Wishart distribution for $p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) = p(\boldsymbol{\mu}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda})$, i.e.,

$$\begin{aligned} p(\boldsymbol{\pi}|\alpha_0) &= \text{Dir}(\{\pi_k\}_{k=1}^K|\alpha_0), \\ p(\boldsymbol{\mu}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda}) &= \prod_{k=1}^K \mathcal{N}(\boldsymbol{\mu}_k|\mathbf{m}_0, (\eta_0 \boldsymbol{\Lambda}_k)^{-1}) \mathcal{W}(\boldsymbol{\Lambda}_k|\mathbf{W}_0, \nu_0), \end{aligned} \tag{3}$$

where $\alpha_0, \eta_0, \mathbf{m}_0, \mathbf{W}_0$ and ν_0 are hyper-parameters. In addition, we assume that $Q(\mathbf{Z}, \boldsymbol{\theta})$ can be factorized as $Q(\mathbf{Z}, \boldsymbol{\theta}) = Q(\mathbf{Z})r(\boldsymbol{\theta})$. Substituting this equation and eq.(2) into eq.(1), we obtain the free energy function of the VB method for Gaussian mixture estimation. In the VB method, the free energy function is maximized with respect to $Q(\mathbf{Z})$ in the VB-E step and maximized with respect to $r(\boldsymbol{\theta})$ in the VB-M step.

2.3 Deterministic Annealing VB Method

The VB enables us to find a local solution because the free energy function of the problem is multimodal. Deterministic annealing is often useful for finding an excellent local solution [7]. Recently, Katahira et al [8] adapted deterministic annealing approach to the VB method, which is called the DAVB method. In the DAVB, the free energy function is modified to fit into the basic equation of the statistical mechanics: $\mathcal{F} = \mathcal{U} - T\mathcal{S}$, where \mathcal{F} is free energy, \mathcal{U} is internal energy, T is temperature and \mathcal{S} is entropy. Using the analogy, the *negative* free energy is represented as

$$\begin{aligned} -\mathcal{F}_\beta(Q(\mathbf{Z}, \boldsymbol{\theta})) &= -\sum_{\mathbf{Z}} \int Q(\mathbf{Z}, \boldsymbol{\theta}) \ln p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta}) d\boldsymbol{\theta} \\ &\quad + \frac{1}{\beta} \sum_{\mathbf{Z}} \int Q(\mathbf{Z}, \boldsymbol{\theta}) \ln Q(\mathbf{Z}, \boldsymbol{\theta}) d\boldsymbol{\theta}, \end{aligned} \tag{5}$$

where β represents the inverse temperature. Note that $\mathcal{F}_\beta(Q(\mathbf{Z}, \boldsymbol{\theta})) = -\mathcal{F}$ in physical meaning, therefore, we refer to $\mathcal{F}_\beta(Q(\mathbf{Z}, \boldsymbol{\theta}))$ as the *negative* free energy. Setting the derivatives of eq. (5) with respect to $Q(\mathbf{Z})$ and $r(\boldsymbol{\theta})$ to 0, we obtain posterior distributions. In the DAVB framework, the free energy function eq. (5) is maximized under various levels of temperature from $\beta = 0$ to $\beta = 1$. Note that the update equations are identical with the standard VB method when $\beta = 1$.

3 Multi-directional Search from the PIP

The DAVB has a potential to find excellent solutions mainly because of the following two reasons. First, it starts searching from the PIP, i.e., the solution at the highest temperature. The underlying idea is that the solution space at the highest temperature would approximate its global structure. Second, it employs annealing, anticipating that a very good local optimum at a certain temperature would be located around a very good local optima at a bit higher temperature. However, the computational cost of the annealing is rather heavy because we must solve the maximization problem many times at various levels of temperature. In this paper, we propose an alternative method to challenge the local optimality problem using only the first property of the DAVB method.

3.1 Primitive Initial Point and Its Properties

Let us call the DAVB solution at the highest temperature ($\beta = 0$) *primitive initial point (PIP)*. We obtain the PIP $\boldsymbol{\theta}^{pip} = \{\alpha^{pip}, \eta^{pip}, \mathbf{m}^{pip}, \mathbf{W}^{pip}, \nu^{pip}\}$ by substituting $\beta = 0$ into the update equations of the DAVB method. Note that, all the mixture components have the identical parameters at the PIP. Using the PIP as the starting point, we can obtain some efficiency in the development of the method. In the next subsection, we will show some empirical evidence that the PIP has some good properties as the starting point for the VB method.

Here, we empirically investigate the properties of the PIP for GMM using 10 real data sets shown in Table 1. Hereafter, we symbolize these data sets as #1,

Table 1. Data set

Data	D	N	Source
australian	11	689	Statlog
bodyfat	14	252	StatLib
breast-cancer	10	683	UCI
diabetes	7	768	UCI
heart	11	270	Statlog
iris	4	137	UCI
liver-disorders	5	345	UCI
mpg	6	383	UCI
space-ga	6	3107	StatLib
wine	13	175	UCI

Table 2. # of eigenvalues and their multiplicity

Data	Negative	Positive
#1	$1^f \times 1, 4^f \times 1, 5^f \times 113, 50^f \times 1$	$5^f \times 11$
#2	$1^f \times 1, 4^f \times 1, 5^f \times 184, 65^f \times 1$	$5^f \times 15$
#3	$1^f \times 1, 4^f \times 1, 5^f \times 93, 45^f \times 1$	$5^f \times 10$
#4	$1^f \times 1, 4^f \times 1, 5^f \times 45, 30^f \times 1$	$5^f \times 7$
#5	$1^f \times 1, 4^f \times 1, 5^f \times 113, 50^f \times 1$	$5^f \times 11$
#6	$1^f \times 1, 4^f \times 1, 5^f \times 14, 15^f \times 1$	$5^f \times 5$
#7	$1^f \times 1, 4^f \times 1, 5^f \times 23, 20^f \times 1$	$5^f \times 5$
#8	$1^f \times 1, 4^f \times 1, 5^f \times 33, 25^f \times 1$	$5^f \times 6$
#9	$1^f \times 1, 4^f \times 1, 5^f \times 32, 25^f \times 1$	$5^f \times 7$
#10	$1^f \times 1, 4^f \times 1, 5^f \times 159, 60^f \times 1$	$5^f \times 13$

#2 \cdots #10. For each data set, each variable was normalized to $[-1, 1]$. We set the number of the components K to be 5 in our experiments.

First, we empirically verify that the PIP forms a saddle point when $\beta = 1$. For each data set, we calculate the Hessian matrix at the PIP and investigate its eigenvalues. The results are shown in Table 2. Note that a superscript f means *-fold*; i.e. $50^f \times 1$ means that the Hessian matrix has one 50-fold eigenvalues. When $\beta = 1$, the Hessian matrix of the negative free energy function at the PIP for each data set has both negative and positive eigenvalues and they have some multiplicities related to the number of components $K = 5$. The results indicate that the PIP actually forms a saddle point of the free energy function when $\beta = 1$, which tells that we need to carefully choose the search directions in order to increase the negative free energy function. In particular, we should perform the searching in the directions in the subspace spanned by the eigenvectors corresponding to positive eigenvalues. In addition, if there are many positive eigenvalues, it is reasonable to choose directions with larger positive eigenvalues. We should search toward eigenvectors corresponding to positive eigenvalues because our aim is to increase the negative free energy function.

Furthermore, all the positive eigenvalues have some degrees of multiplicity related to the number of components K , whose property indicates that only one of the K eigenvectors corresponding to K -fold eigenvalue should be examined, because the multiplicity of eigenvalue represents the redundancy of the mixture component-labeling, and searching in the label-permuted direction only results in label-permuted solutions. We can avoid this redundancy and reduce the computational cost by searching toward only one of multiple eigenvectors.

Next we investigate the negative free energy function along the straight path from the PIP to excellent solutions for all the 10 data sets. The solutions we use here are the best solutions obtained in Experiment I in section 4. Fig. 1 shows the negative free energy function along the straight path from the PIP to the best solution for 8 out of 10 data sets. In eight out of the ten data sets, the negative free energy function monotonically increases along the path. It indicates that, except #1 and #4, there is a path to reach the best DAVB solution just by monotonically increasing the negative free energy function from the PIP. This property indicates that hill climbing from the PIP without annealing might be able to find many (not all) excellent solutions. Note that, even if the negative free energy does not monotonically increase, there might be another good path to an excellent solution.

3.2 Multi-directional Search

Exploiting the properties described above, we propose a multi-directional search algorithm from the PIP. In our approach, we first calculate the PIP estimate $\theta^{pip} = \{\alpha^{pip}, \eta^{pip}, \mathbf{m}^{pip}, \mathbf{W}^{pip}, \nu^{pip}\}$, the Hessian matrix $\partial^2 \mathcal{L} / \partial \theta \partial \theta^T |_{\theta = \theta^{pip}}$ and its eigenvectors and eigenvalues. Second, we select positive eigenvalues whose cumulative contribution ratios are less than 80%. We introduce this selection heuristic in order to select the eigenvectors corresponding to the eigenvalues significantly larger than 0. Then we generate search directions using the selected

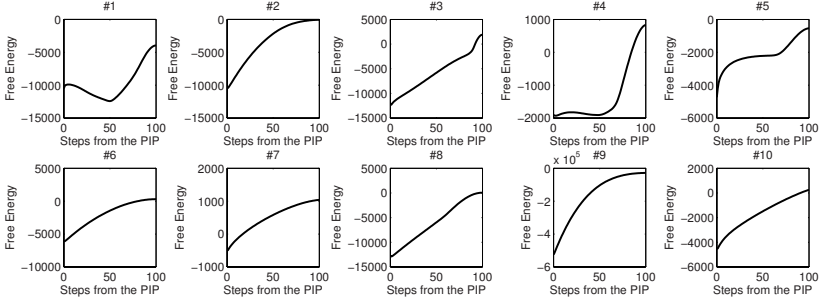


Fig. 1. Cross-section views between the PIP and the best solution

eigenvectors (the details are described later.) Finally, search tokens are generated in the directions around the PIP and perform the VB estimation for each token. We call this PIP-based VB method the VB-PIP method.

The general flow of the VB-PIP is as follows:

VB-PIP method

- 1: Initialize the hyper-parameters of the prior distribution.
- 2: Calculate the PIP estimate $\theta^{pip} = \{\alpha^{pip}, \eta^{pip}, \mathbf{m}^{pip}, \mathbf{W}^{pip}, \nu^{pip}\}$.
- 3: Calculate the Hessian matrix of the negative free energy function eq. (11) at the PIP and perform its eigen-decomposition.
- 4: Select R largest positive eigenvalues according to the contribution ratio.
- 5: Generate search directions using the selected eigenvectors (see below).
- 6: For each direction, iterate the VB-E and VB-M steps until convergence.

In VB-PIP method, we need to specify a set of search directions in the subspace spanned by the selected eigenvectors. We employ the following approach. Let $\mathcal{U} \equiv \{\mathbf{u}_r\}_{r=1}^R$ be the set of normalized selected eigenvectors and $\tilde{\mathcal{U}} \equiv \{-\mathbf{u}_r\}_{r=1}^R$. Let U be the matrix whose r -th column is \mathbf{u}_r , $r = 1, \dots, R$, and an $R \times 2^R$ matrix S whose columns are all possible combinations of the R -dimensional column vector of $+1$ or -1 . Then, compute $V = US/\sqrt{R}$, and let \mathcal{V} denote the set of column vectors of V . In the VB-PIP algorithm, the set of search directions is given by $\mathcal{U} \cup \tilde{\mathcal{U}} \cup \mathcal{V}$. Thus, the VB-PIP algorithm has $2R + 2^R$ search directions in total. We introduced this approach because we want to cover the subspace as uniformly as possible.

4 Performance Evaluation

We evaluate the performance of the VB-PIP using ten real data sets. We compare the VB-PIP with the DAVB and the VB whose initial points are generated by the k -means clustering (VB(kmeans)). In our preliminary experiments (we omit the results here because of space limitation), the DAVB finds good solutions with the small number of initial points, which means many runs are not needed. This fact

Table 3. Performance evaluation

Data	Method	Best	Average	Computational Time			Initial Points	K^*
				Init.	Est.	Total		
#1	VB(kmeans)	-4840	-7082	1.21	298.30	299.51	243	3
	DAVB	-4984	-5889	0.03	293.84	293.87	6	4
	VB-PIP	-4049	-5726	12.42	284.34	296.76	97	3
#2	VB(kmeans)	143	-192	0.42	299.07	299.49	202	3
	DAVB	-65	-149	0.03	287.59	287.62	12	5
	VB-PIP	148	-235	24.68	226.84	251.52	142	4
#3	VB(kmeans)	2342	1970	0.69	298.89	299.58	141	3
	DAVB	1744	1744	0.01	247.36	247.37	3	5
	VB-PIP	2289	1863	4.97	100.27	105.24	42	3
#4	VB(kmeans)	834	427	0.31	296.84	297.15	68	3
	DAVB	834	834	0.01	239.54	239.55	2	3
	VB-PIP	834	747	2.49	296.74	299.23	71	3
#5	VB(kmeans)	-575	-1786	0.77	298.96	299.73	479	1
	DAVB	-537	-1099	0.04	298.15	298.19	28	2
	VB-PIP	-447	-730	6.36	70.46	76.82	76	2
#6	VB(kmeans)	319	293	0.65	289.10	289.75	500	5
	DAVB	301	244	0.04	294.45	294.49	31	5
	VB-PIP	319	274	0.06	6.59	6.65	14	5
#7	VB(kmeans)	1092	1079	0.33	298.08	298.41	151	3
	DAVB	1039	1039	0.01	250.14	250.15	5	3
	VB-PIP	1092	1062	0.39	64.19	64.58	24	3
#8	VB(kmeans)	42	-173	0.66	297.75	298.41	301	4
	DAVB	42	42	0.03	281.74	281.77	13	4
	VB-PIP	54	-27	0.42	32.49	32.91	24	5
#9	VB(kmeans)	-28032	-28088	0.21	288.57	288.78	13	5
	DAVB	-28139	-28139	0.05	248.23	248.28	3	5
	VB-PIP	-28032	-28099	1.96	239.24	241.20	14	5
#10	VB(kmeans)	288	59	0.55	292.62	293.17	500	2
	DAVB	252	-39	0.07	295.89	295.96	60	2
	VB-PIP	288	-13	17.66	159.19	176.85	272	2

is actually an advantage of the DAVB, that is, the DAVB finds good solutions more stably than the others. Considering this property of the DAVB, we compare the performances of three methods limiting the computational time rather than running them with the same number of initial points. In each experiment, the hyper-parameters $\{\alpha_0, \eta_0, \mathbf{m}_0, \mathbf{W}_0, \nu_0\}$ for the prior distributions eqs. (3) and (4) are set to be $\{1, 1, \bar{\mathbf{x}}, 10 \times I, 50\}$, where $\bar{\mathbf{x}}$ is the mean vector. The temperature scheduling for the DAVB is set to be $\beta^{(0)} = 0.1$, and $\beta^{(t+1)} = \beta^{(t)} \times 1.2$.

Here, we limit the maximum computational time to be 300 seconds and compare the best solutions obtained within the time. The results are shown in Table 3. The quality of solutions is measured by the negative free energy; the larger, the better. In the table, ‘‘Init.’’ indicates computational time spent to generate initial points, while ‘‘Est.’’ denotes that spent to estimate parameters,

and “Total” = “Init.” + “Est.”. Moreover, “Initial Points” and “ K^* ” means the number of initial points and the estimated number of components, respectively.

The VB-PIP finds the best solutions in nine out of ten data sets; it finds the strictly best in four data sets, and finds the tying best in five data sets. In addition, the VB-PIP finds those excellent solutions with the smaller number of initial points compared with the VB(kmeans), which indicates the VB-PIP performs more stably than the VB(kmeans). We think this is because the VB-PIP inherits the stability of the DAVB even without annealing process. This result indicates that the initial directions generated by the VB-PIP have significant influence over obtaining excellent solutions.

5 Conclusion

In this paper, we defined the primitive initial point (PIP) for the VB framework using the DAVB method. We empirically examined the properties of the PIP. Exploiting the results, we proposed a multi-directional search VB method with the PIP as its starting point. In our experiments, the proposed method showed performance comparable with or better than the other two VB methods. In the future, we need to further investigate the properties of the PIP in order to understand the rationale for efficiency of generating initial directions starting from the PIP.

References

1. McLachlan, G.J., Peel, D.: Finite mixture models. John Wiley & Sons, Chichester (2000)
2. Attias, H.: Inferring parameters and structure of latent variable models by variational bayes. In: Proc. of the 15th Conf. on Uncertainty in Artificial Intelligence, pp. 21–30 (1999)
3. Ueda, N., Ghahramani, Z.: Bayesian model search for mixture models based on optimizing variational bounds. *Neural Networks* 15(10), 1223–1241 (2002)
4. Saul, L.K., Jaakkola, T., Jordan, M.I.: Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research* 4, 61–76 (1996)
5. Opper, M., Saad, D. (eds.): Advanced mean field method: theory and practice. MIT Press, Cambridge (2001)
6. Corduneanu, A., Bishop, C.M.: Variational Bayesian model Selection for mixture distributions. In: Proc. of 8th Int. Conf. on Artificial Intelligence and Statistics, pp. 27–34 (2001)
7. Ueda, N., Nakano, R.: Deterministic annealing EM algorithm. *Neural Networks* 11(2), 271–282 (1998)
8. Katahira, K., Watanabe, K., Okada, M.: Deterministic annealing variant of variational Bayes method. In: Proc. of Int. Workshop on Statistical-Mechanical Informatics, pp. 65–73 (2007)
9. Ishikawa, Y., Nakano, R.: Landscape of a likelihood surface for a Gaussian mixture and its use for the EM algorithm. In: Proc. of the Int. Joint Conf. on Neural Networks, pp. 1434–1440 (2006)
10. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)

A Bayesian Graph Clustering Approach Using the Prior Based on Degree Distribution

Naoyuki Harada¹, Yuta Ishikawa¹, Ichiro Takeuchi¹, and Ryohei Nakano²

¹ Nagoya Institute of Technology

Gokiso-cho, Showa-ku, Nagoya, 466-8555, Japan

{harada,ishikawa}@goat.ics.nitech.ac.jp, takeuchi.ichiro@nitech.ac.jp

² Chubu University

1200 Matsumoto-cho, Kasugai-shi, Aichi, 486-8507 Japan

nakano@cs.chubu.ac.jp

Abstract. Newman et al. proposed a stochastic graph clustering approach using a mixture model with an assumption that a group of vertices is regarded as a class when the vertices have a similar connection pattern. Kuwata et al. recently adopted a nonparametric Bayesian approach and improved Newman’s one in such a way that the number of classes can also be empirically estimated. In this paper, we propose a new approach that can incorporate the degree distribution of the network structure as priors for Bayesian estimation. We show the effectiveness of our method through experiments using both artificial and real data.

Keywords: complex networks, graph clustering, variational Bayes method.

1 Introduction

Network structures observed in a variety of fields had been independently studied in each literature. Recently, however, the existence of common structural feature was revealed in many different types of networks such as the World Wide Web, co-authorship of scientific papers and protein interaction networks [1, 2]. Studying the common structures in various networks is now recognized as an important step to understand many complex real-world problems, and a term “complex network” is often used to represent the research area. Graph clustering is one of the effective ways to analyze complex networks [3, 4]. By clustering a network based on its connecting patterns, a group structure of the network may be identified. In this paper, we are concerned with the problem of graph clustering.

Newman et al. proposed a stochastic graph clustering method using a mixture model with an assumption that a group of vertices is regarded as a class when the vertices have a similar connection pattern [5]. One practical difficulty in this method is that we have to provide the true or appropriate number of classes which is hardly known in general. To solve this problem, Kuwata et al. recently adopted a nonparametric Bayesian approach and improved Newman’s method

in such a way that the number of classes can also be empirically estimated [6]. This nonparametric Bayesian approach works well if we can specify appropriate prior distributions of the model parameters. On the other hand, if the provided prior distributions are not appropriate, the estimation would fail.

In this paper, we propose to use a degree distribution of the network for specification of the prior distributions in nonparametric Bayesian estimation. The degree distribution is a histogram of the number of edges that each vertex has (i.e., degree) and it is important prior information about the existence of edges (See Figs 1 and 2 for two examples of degree distributions). In the literature of complex networks, various types of degree distributions are observed and considered since the degree distribution has important information to characterize the network property. We first provide a simple alternative model for graph clustering and then suggests a heuristic to incorporate the degree distribution of the network into the prior distributions. We demonstrate the effectiveness of our approach through artificial and real data experiments.

In Section 2, we briefly review the related studies. In Section 3, we describe our proposed method. Then, in Section 4, we evaluate our method using computer generated test networks and a real network. Finally, we close in Section 5 with concluding remarks.

2 Background

2.1 Graph Clustering Problem

Suppose we have a network with N vertices connected by directed edges. The network can be represented as an adjacency matrix A where A_{ij} is 1 if there is an edge from vertex i to j and 0 otherwise. This $N \times N$ matrix is observable. Suppose also that C is the number of classes or groups in the network and g_i denotes the group which vertex i belongs to. The graph clustering is defined as the problem of estimating the latent group memberships g_i from the observable adjacency matrix A . In stochastic approaches, we estimate $q_{ir} = P(g_i = r|A)$, $i = 1, \dots, N$, $r = 1, \dots, C$, the probability that the vertex i belongs to group r given the adjacency matrix A . Note that C is generally unknown and should also be estimated.

2.2 Related Works

Newman et al. proposed a probabilistic mixture model [5]. They defined two kinds of parameters θ_{rj} and π_r . θ_{rj} is the probability that a directed edge from a particular vertex in group r connects to vertex j and satisfies $\sum_{j=1}^N \theta_{rj} = 1$. π_r is the fraction of vertices in group r and satisfies $\sum_{r=1}^C \pi_r = 1$. Using these parameters, the likelihood is defined as

$$P(A, g|\pi, \theta) = \prod_{i=1}^N \left[\pi_{g_i} \prod_{j=1}^N \theta_{g_i, j}^{A_{ij}} \right]. \quad (1)$$

Using the EM algorithm, the likelihood (II) is maximized with respect to the probability of group memberships q_{ir} and parameters $\{\theta_{rj}, \pi_r\}$. In this method, we need to manually specify the appropriate number of classes C .

Kuwata et al. [6] proposed to use a nonparametric Bayesian approach for Newman’s probabilistic mixture model. This approach enables us to estimate the number of classes C . They implemented the approach using the Stick-breaking representation of Dirichlet process. The parameters in Newman’s model are replaced as

$$P(\theta) = \prod_{r=1}^{\infty} Dirichlet(\theta_{r1}, \dots, \theta_{rN} : \phi_{r1}, \dots, \phi_{rN}) \tag{2}$$

$$\pi_{g_i} = P(g_i | v_1, \dots, v_{\infty}) = v_{g_i} \prod_{k=1}^{g_i-1} (1 - v_k) \tag{3}$$

where *Dirichlet* is a Dirichlet distribution and all v_r are generated from a beta distribution $Beta(v_r | 1, \alpha)$. And the parameter α follows a gamma distribution $Gamma(\alpha; c_1, c_2)$ where $\{c_1, c_2\}$ is a set of constant values ($\{c_1, c_2\} = \{1.0, 1.0\}$ in the experiments described later). The probability of group memberships q_{ir} and approximate posteriors of parameters are calculated by the variational Bayes method.

2.3 Variational Bayes Method

The variational Bayes method is an approximation of the Bayes method. In Variational Bayesian framework, the lower bound of marginal log likelihood $\ln P(D)$ is maximized. The lower bound $F[Q]$ called variational free energy is derived as

$$\begin{aligned} \ln P(D) &= \log \sum_Z \int P(D, Z, \Theta) d\Theta \\ &= \log \sum_Z \int Q(Z, \Theta) \frac{P(D, Z, \Theta)}{Q(Z, \Theta)} d\Theta \\ &\geq \sum_Z \int Q(Z, \Theta) \ln \frac{P(D, Z, \Theta)}{Q(Z, \Theta)} d\Theta \equiv F[Q] \end{aligned} \tag{4}$$

where D denotes the observable data set, Z is the latent variables, Θ is the parameters and $Q(Z, \Theta)$ is an arbitrary distribution called the variational posterior distribution. The relationship between $\ln P(D)$ and $F[Q]$ is also expressed as

$$\ln P(D) = F[Q] + KL[Q(Z, \theta) || P(Z, \theta | D)]. \tag{5}$$

From this equation, since $\ln P(D)$ is constant under the given observable data D , maximizing the lower bound $F[Q]$ corresponds to minimizing the KL divergence $KL[Q || P]$. Hence, Q obtained after maximizing $F[Q]$ is the best approximation of

the true posterior $P(Z, \Theta|D)$. As we noted above, Q is an arbitrary distribution, so we assume that it can be factorized as

$$Q(Z, \Theta) = Q(Z)Q(\Theta). \quad (6)$$

In many cases, the joint distribution $P(D, Z, \Theta)$ can be decomposed as

$$P(D, Z, \Theta) = P(D|Z, \Theta)P(Z)P(\Theta). \quad (7)$$

Substituting Eq. (6) and Eq. (7) into Eq. (4), we obtain $F[Q]$. The variational Bayes method repeats, what is called, VB-E step and VB-M step. In VB-E step, $F[Q]$ is maximized with respect to $Q(Z)$, and in VB-M step, $F[Q]$ is maximized with respect to $Q(\Theta)$.

3 The Proposed Method

The performance of Bayesian estimation highly depends on how to specify the prior distributions. Since the optimal priors are unknown in general, conjugate priors are usually used for its mathematical convenience. In addition, there are no systematic way to specify the hyper-parameters of the prior distributions, although they have a large influence on the estimation. In this paper, we propose a heuristic that enables us to use the degree distribution of the network for the specification of more appropriate priors in nonparametric Bayesian graph clustering.

3.1 Alternative Model for Graph Clustering

To incorporate the proposed heuristic, we consider an alternative model for graph clustering. Let us define E_{rj} as the probability that a directed edge exists from a vertex in group r to a vertex j . Unlike θ_{rj} in [5], we assume that E_{rj} are independent each other. The distribution of A given g and E is the product of binomial distributions and defined as

$$P(A|g, E) = \prod_{i=1}^N \prod_{j=1}^N E_{g_i j}^{A_{ij}} (1 - E_{g_i j})^{1 - A_{ij}}. \quad (8)$$

To incorporate nonparametric Bayesian approach in this model, we set the prior distribution of E as

$$P(E) = \prod_{r=1}^T \prod_{j=1}^N \text{Beta}(E_{rj}; \phi_{rj1}, \phi_{rj2}). \quad (9)$$

Note that a beta distribution is the conjugate prior of a binomial distribution. The joint distribution of our model is decomposed as

$$P(A, g, V, E, \alpha) = P(A|g, E)P(g|V)P(V|\alpha)P(\alpha)P(E). \quad (10)$$

Applying the variational Bayes method to this model, we can obtain a good approximation of $P(g, V, E, \alpha|A)$ which includes the distribution of group memberships for each vertex.

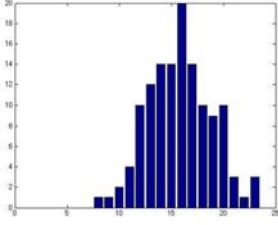


Fig. 1. A single-peaked degree distribution

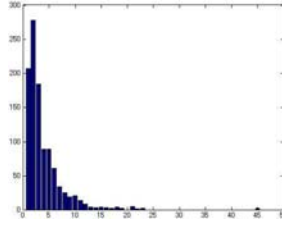


Fig. 2. A long-tailed distribution

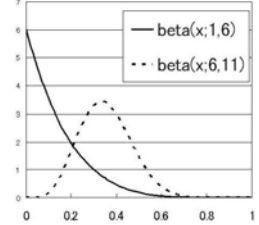


Fig. 3. Various shapes of beta distribution

3.2 Setting of Priors Based on Degree Distribution

The advantage of the model [\(8\)](#) is that it is easy to introduce the degree distribution of the network into the prior distributions of the model parameters. In particular, the degree distribution of the network can be directly incorporated to the hyper-parameters of the prior distributions $P(E)$. We divide the degree distributions into two patterns by its shape, *single-peaked* and *long-tailed*. In this paper, we use the term “single-peaked” for a distribution which has a peak (occasionally peaks) such as Poisson distributions in random networks ([Fig. 1](#)). In contrast, we use the word “long-tailed” for a distribution which has a long tail such as power-law distributions and exponential distributions that is often observed in complex networks ([Fig. 2](#)). Since the prior distribution $P(E_{rj})$ is modeled by a beta distribution, its shape is controlled by the hyper-parameter ϕ (See [Fig. 3](#) for two examples of beta distribution with different ϕ). By adjusting ϕ , we can approximate the shape of $P(E)$ close to the degree distribution. In particular, for the single-peaked degree distribution, we set the prior distribution as

$$P(E_{rj}; \phi_{rj1}, \phi_{rj2}) = \text{Beta} \left(E_{rj}; 1 + \frac{k_j}{T}, 1 + \frac{N - k_j}{T} \right). \quad (11)$$

For the long-tailed degree distribution, we set the prior distribution as

$$P(E_{rj}; \phi_{rj1}, \phi_{rj2}) = \text{Beta} \left(E_{rj}; 1, \frac{N - k_j}{k_j} \right), \quad (12)$$

Here, k_j is the number of incoming edges of vertex j and T is the maximum number of classes we have to set in advance. In the experiments described later, we use large enough T so that its choice does not affect the final model.

4 Experiments

We examine the performance of our method using three kinds of networks: two artificial networks and one real network. Artificial networks in experiment 1 show assortative mixing. In experiment 2, networks also show disassortative

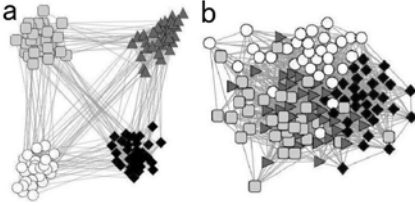


Fig. 4. Benchmark of Girvan and Newman: Each network corresponds to the conditions, (a) $Z_{out}=2$ and (b) $Z_{out}=8$. These networks show assortative mixing. In case of $Z_{out}=8$, the four groups are almost mixing each other and difficult to be divided into.

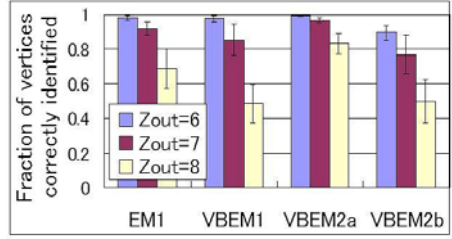


Fig. 5. The comparison result of the computer generated benchmark: Our proposed method VBEM2a which has suitable prior distributions for the networks shown in Fig. 4 could correctly classify more than 80% vertices even in the case of $Z_{out}=8$

mixing besides assortative mixing. Both of two kinds of networks have single-peaked degree distributions. In experiment 3, we deal with a real network with a long-tailed degree distribution. In these experiments, “EM1” and “VBEM1” represent the methods proposed in [5] and [6] respectively, while “VBEM2a” and “VBEM2b” are our methods. In VBEM2a, the hyper-parameters which control the prior distributions are adjusted to a single-peaked degree distribution. In contrast, the hyper-parameters of VBEM2b are adjusted to a long-tailed degree distribution. The maximum number of classes T is set to 20 in experiments 1 and 2, and 100 in experiment 3.

4.1 Experiment 1

First, we focus on the networks with assortative mixing in which vertices divide into groups such that the members of each group are mostly connected to other members of the same group (also called “community structure”). We adopt one of the most famous computer generated benchmarks designed by Girvan and Newman [7] to evaluate the performance of each method. Each graph consists of 128 vertices and 4 equivalent size groups. The average degree is set to 16. A parameter Z_{out} denotes the average number of edges that a vertex connects to the vertices in different groups (Fig. 4). A typical degree distribution is shown in Fig. 1. In this experiment, we consider three situations $Z_{out}=6, 7$ and 8 . 10 networks are generated under the same value of Z_{out} and all four methods are applied 30 times with different initial values to each network. We collect the best score (the fraction of vertices correctly identified: FCI) among the 30 estimations for each network. The results are shown in Fig. 5 where each bar and a line across the bar represent the mean and standard deviation of the best FCIs among 10 samples respectively.

Our proposed method VBEM2a shows better result than EM1 in spite of estimating the same things as VBEM1, while VBEM1 is not as precise as its

Table 1. A class-wise probability matrix $C^* = 5$ (e.g., the element (C_1, C_2) is the probability that a vertex in C_1 attaches an edge to another vertex in C_2)

	C_1	C_2	C_3	C_4	C_5
C_1	0.33	0.33	0.32	0.01	0.01
C_2	0.48	0.01	0.01	0.01	0.49
C_3	0.33	0.01	0.33	0.32	0.01
C_4	0.01	0.01	0.33	0.33	0.32
C_5	0.01	0.48	0.01	0.49	0.01

Table 2. The result of clustering $C^* = 5$ artificial networks: Each element shows mean and standard deviation of 100 samples (the best score among 10 estimations for each network)

	FCI	estimated C
EM1	0.872 ± 0.082	5.00 (fixed)
VBEM1	0.933 ± 0.110	5.54 ± 1.06
VBEM2a	0.942 ± 0.094	5.07 ± 0.41
VBEM2b	0.854 ± 0.079	8.82 ± 1.93

original method EM1 because of estimating the number of classes. The point to be focused on is not only its high mean value but also its small standard deviation which means this method is robust against the subtle difference of networks.

4.2 Experiment 2

In experiment 2, we compare the four methods using artificial networks which have more complex structure than what we considered in experiment 1: mixture of assortative and disassortative (opposite to “assortative”), i.e., a number of edges are generated among the vertices in different groups as well. 100 networks with the same number of vertices and average degree as experiment 1 are generated. They contain five classes of almost equal size. Directed edges are generated according to the probability in Table 1. Even in this case, since the vertices in the same group have a similar connecting pattern, it is possible to discover the original groups in the networks. The results are shown in Table 2.

Again, VBEM2a shows the best solutions from the both viewpoints of FCI and the estimated number of classes. In contrast to the result in experiment 1, VBEM1 shows better result than EM1 even though the true number of classes is given to the latter. From these results, we conjecture that the nonparametric Bayesian approach enable us not only to estimate the appropriate number of classes, but also to find out complex structures in networks.

4.3 Experiment 3

Finally, we evaluate the performance of our methods using a real data set: co-authorship of papers adopted in NIPS [8]. We picked up the biggest component (a connected part of network) of the data which contains 1061 vertices and 2080 undirected edges. The degree distribution of this network is shown in Fig. 2. Different from artificial networks, real networks don’t provide us with the correct answer (the true memberships of each vertex). Therefore we show the result of clustering as clustered adjacency matrix (Fig. 6).

VBEM2b found a meaningful group structure from the network thanks to the appropriate prior setting. On the other hand, VBEM2a which showed the best

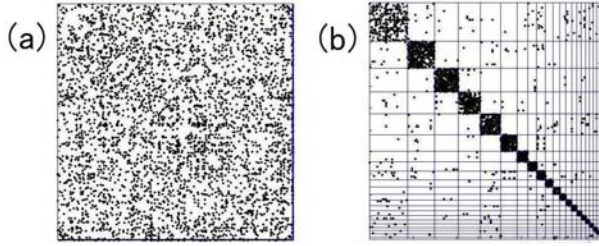


Fig. 6. The clustered adjacency matrices with the highest $F[Q]$ (See section 2) in 50 times application of each method: A black point corresponds to the element $A_{ij} = 1$. (a) VBEM2a found 3 groups structure but it seems meaningless. On the other hand, (b) VBEM2b which has suitable setting prior distributions found 22 group structure and it shows strong assortative mixing.

performance in the previous two experiments did not work well because of the mismatch between its prior distributions and the network's degree distribution. This result tells us that we should carefully consider the setting of the prior distributions to obtain good performance.

5 Conclusion

In this paper, we proposed a graph clustering method that can incorporate the degree distribution of the network. Numerical experiments demonstrate that our method can work better than conventional methods. As a future work, we plan to investigate the applicability of our approach to wider range of real-world networks.

References

1. Barabasi, A.-L.: *Linked: The New Science of Networks*. Perseus Books, Cambridge (2002)
2. Newman, M.E.J.: The structure and function of complex networks. *SIAM Review* 45, 167–256 (2003)
3. Danon, L., Duch, J., Diaz-Guilera, A., Arenas, A.: Comparing community structure identification. *J. Stat. Mech.*, P09008 (2005)
4. Fortunato, S., Castellano, C.: Community structure in graphs. In: *Encyclopedia of Complexity and System Science*. Springer, Heidelberg (2009)
5. Newman, M.E.J., Leicht, E.A.: Mixture models and exploratory data analysis in networks. *Proc. Natl. Acad. Sci. USA* 104, 9564–9569 (2007)
6. Kuwata, S., Ueda, N., Yamada, T.: Graph Clustering based on the Nonparametric Bayes Model. *IEICE Technical Report*, vol. 107, no. 115, PRMU2007-41, pp. 81–86 (2007) (in Japanese)
7. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* 99, 7821–7826 (2002)
8. Roweis, S.: Data for MATLAB hackers NIPS Conference Papers., <http://www.cs.toronto.edu/~roweis/data.html>

Common Neighborhood Sub-graph Density as a Similarity Measure for Community Detection

Yoonseop Kang and Seungjin Choi

Department of Computer Science
Pohang University of Science and Technology
San 31 Hyoja-dong, Nam-gu, Pohang 790-784, Korea
{e0en, seungjin}@postech.ac.kr

Abstract. Community detection in networks involves grouping nodes on a graph into clusters such that connections between groups are sparse while nodes within groups are densely connected. Despite the success of clustering based community detection methods, there have been few efforts to devise similarity metrics between nodes for clustering algorithms that measures the likeliness of two nodes belonging to the same community. In this paper we present a new similarity measure based on the density of a sub-graph constructed by common neighbors of two nodes in question. The proposed metric is referred to as *common neighborhood sub-graph density* (CND) and is combined with affinity propagation to detect communities from network data. We apply community detection algorithms with CND to real-world benchmark data sets to demonstrate its useful behavior in the task of community detection in networks.

1 Introduction

Complex systems take the form of networks, where relationships among a set of entities are represented by graphs. Exemplary systems include living organisms, ecosystems, economy, world wide web, and social networks.

One of important issues in understanding networked data is the detection and characterization of *community structure* in networks. Community detection in networks is to group nodes in a graph into clusters within which the network connections are dense but between which they are sparser [9].

Various approaches for community detection including graph partitioning and clustering have been developed [2]. Due to the similarity between community detection and clustering problem, attempts to tackle the community detection problem using pre-existing clustering algorithms arouse naturally [5]. The key idea of converting community detection into clustering problem is to define a similarity measure that effectively reflects connection between nodes in a network. Although there are many similarity measures that are already used in the area of graph theory, there has been few similarity measure that is specifically designed for the task of community detection.

In this paper we present a new similarity measure which is designed for the task of community detection, referred to as *common neighborhood sub-graph density* (CND). CND aims to directly estimate whether two nodes are in a same

community or not by examining the connection between the common neighbors of them. As a clustering method, we used affinity propagation [3]. We investigate the relationship of CND and affinity propagation, and investigate the effectiveness of the combination of CND and affinity propagation through experiments on real-world network data.

2 Proposed Similarity Measure

If we know the community that a node belongs to, it would be ideal to give infinite similarity for nodes from the same community and zero for the nodes from different communities. Although it is obviously not possible in real problems, we can try to predict if a pair of nodes belongs to the same community or not.

To define an appropriate similarity measure that indicates that a pair belongs to a same community or not, we start from an insight about communities in a social network. A community is defined as a group of nodes with dense connections within the group. It means that we will have dense edge distributions among any set of nodes from a same community.



(a) Two nodes from different communities connected by an edge (b) Two nodes from different communities share many common neighbors

Fig. 1. Cases that direct connection between nodes and Jaccard's index fails as an indicator of two nodes denoted as black circles belonging to same community. Common neighbor nodes are denoted as grey circles.

It is possible that a pair of nodes being connected by an edge even if they belong to different communities (Fig. 1(a)). However, it is unlikely that two nodes from different communities have many common neighbors. It is because if two nodes belong to the different communities, then the nodes would be mostly connected to nodes from their own communities.

We go one step further from this idea. If the edges of two nodes are spreaded to several communities, they may share many common neighbors even if they are from different communities (Fig. 1(b)). As the connection among nodes from different communities are sparse according to definition of a community, the connection among such common neighbors will still be sparse if they are from several different communities. In the other hand, the connection among the common neighbors will be dense if they are from the same community.

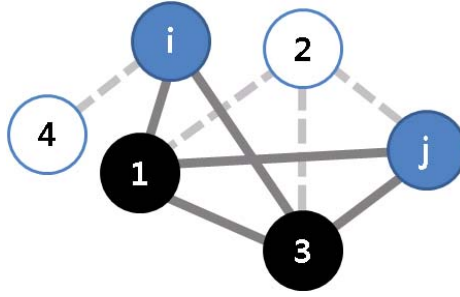


Fig. 2. Example graph for description of CND. Given node i and j , the common neighbor \mathcal{C}_{ij} is $\{i, j, 1, 3\}$ and the value of CND K_{ij} is sum of the weights of the edges that are depicted as solid lines on figure. Note that K_{ij} has nonzero value even though edge connecting node i and j does not exist.

Therefore, density of edges connecting the common neighbors of two nodes is better choice than the number of common neighbors of them for community detection task.

Formally, the edge density K_{ij} of a subgraph formed by common neighbors of two nodes i and j is defined as the ratio of the number of existing edges between nodes and maximum possible number of edges between nodes (Fig. 2):

$$K_{ij} = \frac{\sum_{k \in \mathcal{C}_{ij}} \sum_{l \in \mathcal{C}_{ij}} A_{kl}}{|\mathcal{C}_{ij}|(|\mathcal{C}_{ij}| - 1)}, \quad (1)$$

where \mathcal{C}_{ij} is index set of common neighbors of node i and j :

$$\mathcal{C}_{ij} = \{i, j\} \cup \{k | A_{ik} \neq 0 \text{ and } A_{jk} \neq 0\}, \quad (2)$$

and adjacency matrix \mathbf{A} is defined to be a matrix which has nonzero elements when there exist an edge connecting two nodes. Although the relation between nodes can be asymmetric, we only focus on the symmetric case (i.e. $A_{ij} = A_{ji}$).

The sub-graph density K_{ij} is sensitive to the existence of noisy edges. If the number of common neighbors of two node is small, the effect of noisy connection between the common neighbors significantly affects K_{ij} . For example, if two nodes are connected with one common neighbor with noisy edges, similarity measure K between them never falls below $2/3$.

Therefore, we modify K to give higher similarity for nodes with bigger number of common neighbors and less similarity for nodes with small number of common neighbors to suppress the effect of noisy connections. This can be done by simply multiplying $|\mathcal{C}_{ij}|^\gamma$. The resulting definition of our proposed similarity measure, CND becomes as below:

$$K_{ij} = |\mathcal{C}_{ij}|^\gamma \frac{\sum_{k \in \mathcal{C}_{ij}} \sum_{l \in \mathcal{C}_{ij}} A_{kl}}{|\mathcal{C}_{ij}|(|\mathcal{C}_{ij}| - 1)}, \quad (3)$$

where γ is a constant parameter. Greater value of γ gives higher similarity to the nodes with more common neighbors.

3 Comparison to Existing Similarity Measures

Independent to the problem of community detection, many measures of 'distance' between nodes has developed in the field of graph theory: including geodesic distance, resistance distance [6], and Jaccard's index.

Similar to CND, Jaccard's index also considers common neighbors of node i and j for calculating similarity. Jaccard's index $J(i, j)$ is the number of common neighbors of node i and j divided by the union set of neighbors of the two nodes. Using Jaccard's index for community detection has been shown to be effective in detecting communities from complex networks than competing algorithms [10].

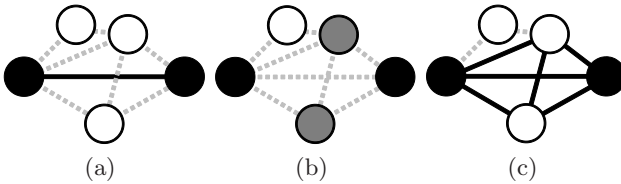


Fig. 3. Difference among similarity measures defined for two nodes (notated as black filled circles) in a graph. edge weight (a), Jaccard's index (b), and CND (c). Dark-colored edges or nodes are taken account when calculating similarity measures.

Although Jaccard's index and CND looks similar, the critical difference between is that Jaccard's index only takes care about the common neighbor nodes and CND considers the connectivity among them (Fig. 3).

4 Combination with Affinity Propagation

To detect communities in a graph using a similarity measure, one needs a clustering algorithm to partition a graph according to the given similarity measure. The most critical problem of well-known clustering algorithms including k-means and hierarchical clustering is that the number of cluster must be given by user. *Affinity propagation* is a clustering algorithm that automatically decides number of clusters [3].

Basically, affinity propagation finds clusters by identifying data points called *exemplars* that represent their own clusters. The algorithm is basically the procedure of determining exemplars and points that are in the same clusters of the exemplars. In determining exemplars, the input preference $s(k, k)$ for each data points plays an important role. Higher value of input preference $s(k, k)$ for a data point k gives higher likelihood of a point k being selected as an exemplar of a cluster.

Therefore, the number and accuracy of clusters detected by affinity propagation is strongly affected by the choice of input preference of each nodes $s(k, k)$, which represents the likelihood of k th node being an exemplar of a cluster. Giving

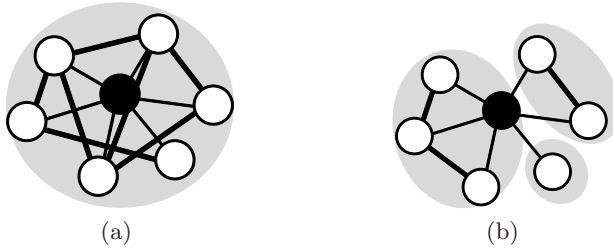


Fig. 4. Comparison between a node’s degree and its self-CND as a input preference. Each oval-shaped shaded area means a natural community. (a) is more likely to be an exemplar of a community than (b).

higher values for $s(k, k)$ results into more number of clusters. Although setting $s(k, k)$ as a constant value for all nodes k is a fairly working heuristic, there is a room for further improvement.

The most straightforward improvement would be choosing a degree of a node as its input preference. However, this may not work on some situations. If a node is connected to many communities, its neighbors will be sparsely connected to each other. If two nodes have same number of neighbors and only densities of connection among their neighbors are different, using degree of a node will not be able to catch the difference of two nodes (Fig. 4). It would be better to give higher input preference to a node whose neighbors are more densely connected to each other, because it means the node is more likely to be connected to a single community and therefore more preferable as a exemplar of a community than other one. CND provides us a nice way of detecting nodes that are highly connected to small number of communities.

When we recall the definition of a community - a group of nodes densely connected to each other, it will be safe to assume that if there are dense connection between the a set of nodes, then they are members of a same community. K_{ii} measures the density of connections among the neighbors of node i and itself. If K_{ii} is low, we can assume that its neighbors are spreaded to multiple communities, or it simply has small number of neighbors, which in both of the cases the node is not suitable as an exemplar.

5 Experiments

5.1 Effect of Parameter γ on Noisy Networks

In definition of CND, we introduced a term $|\mathcal{C}_{ij}|^\gamma$ to prevent giving high similarity values for node pairs from different communities that share small number of common neighbors in noisy networks. To examine the effect of the introduced term, we tested our method by varying parameter γ of CND in synthetically created noisy network data.

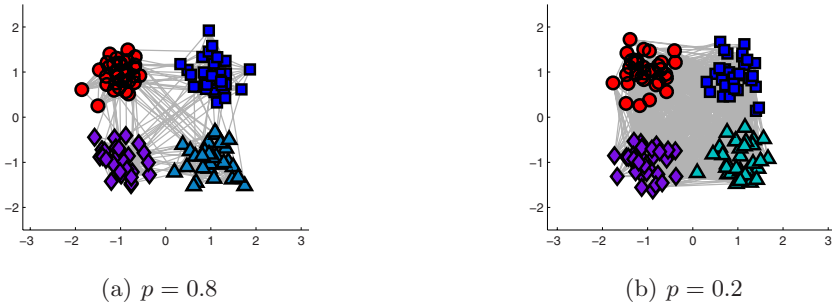


Fig. 5. Pictorial description of noisy 4-community network data with different value of p

As a noisy network data, we created a network data with 128 nodes and divided them into 4 equal communities with 32 nodes. After creating nodes, we randomly added the links connecting nodes. Every node was connected to 32 other nodes in average.

Among N links allocated for a node, we added pN links to connect a node to its own community and $(1 - p)N$ nodes to connect it to different communities ($0 \leq p \leq 1$). By varying p we could control the ratio of links connecting different communities. In other words, the level of noise in network data. Larger value of p leads to less connection among communities (Fig. 5).

We created three random networks with different noise levels by setting p to 0.2, 0.5, and 0.8. To see the effect of parameter γ of CNM on the accuracy of community detection, we varied γ from 0.1 to 3.0.

The accuracy of detecting communities from network was measured in terms of average purity of detected communities, which is defined as

$$\frac{1}{K} \sum_k \sum_{i \neq j \text{ and } i, j \in C_k} \delta(i, j) / \sum_k |C_k|^2, \tag{4}$$

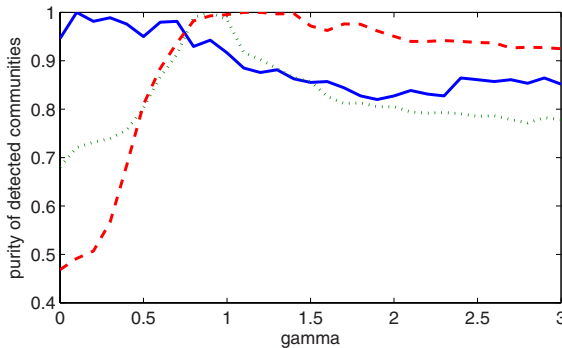


Fig. 6. Effect of parameter γ on noisy 4-community network data. (Blue solid line: $p = 0.8$, green dotted line: $p = 0.5$, red dashed line: $p = 0.2$).

where K is number of detected communities, $\delta(i, j) = 1$ if node i and j belongs to the same ground-truth community and also same detected community, and $|\mathcal{C}_k|$ is a size of k th detected community [10].

Every choice of γ gave the correct number of community. However, the purity of detected communities was differed by change on γ and noise level of network. When noise level of the network was relatively low, the purity of detected communities slightly decreased as γ increased. However, when level of noise get higher, purity of detected communities increased as value of γ increased, until the value reached the optimal value, which was around 1.0 (Fig. 6). When γ grew larger than the optimal value, purity of detected communities also decreased. From this result, we can conclude that parameter γ becomes more important as the noise level of increases and larger value of γ gives more correct detection result than small value of γ .

However, as purity decreased as γ grew larger than the optimal value, finding the optimal value of γ becomes important. We decided to use *modularity* [9] Q as a measure for finding optimal value of γ . Modularity is a widely used performance measure of community detection methods which measures the difference between number of links within a detected communities and links within a random graph with equal degrees for each nodes. Given an affinity matrix \mathbf{A} and detected communities, modularity is defined as:

$$Q = \frac{1}{4m} \sum_i \sum_j s(i, j) \left(\mathbf{A}_{ij} - \frac{k_i k_j}{2m} \right), \quad (5)$$

where \mathbf{A} is an affinity matrix, m is a total number of links, k_i is degree of node i , and $s(i, j) = 1$ if node i and j belongs to a same detected community, and -1 otherwise. On further experiments, γ was set to a value that gives highest modularity of detected communities.

5.2 Input Preference for Affinity Propagation

We compared various choice of input preferences $s(i, i)$, which includes constant value (CND-const), degree of nodes (CND-degree), and diagonal elements $K(i, i)$ of CND (CND-autosim). As a constant input preference, we used a median value of entries in similarity matrix, which is a nice-working choice. Finally, as affinity propagation takes the negative similarity values, we subtracted the maximum value of similarity values multiplied by 2.5 as below:

$$s(i, i) := s(i, i) - 2.5 \max_{i, j} (s(i, j)). \quad (6)$$

We calculated the number of detected communities and purity of the detected communities detection methods.

For every data set, using diagonal elements of CND gave the most correct number of communities to show that it was the best choice for input preference for affinity propagation (Table 1). Although using constant value (CND-const)

Table 1. Performance comparison of input preference measures for affinity propagation

Data Set (# communities)		CND-const	CND-degree	CND-autosim
College Football(12)	# of communities	15	11	11
	purity	0.8507	0.8368	0.8529
Political Blogs(3)	# of communities	12	2	2
	purity	0.8397	0.8215	0.8238
Political Books(2)	# of communities	7	4	3
	purity	0.7136	0.6768	0.7098

got the highest purity on two of three data sets, it failed to detect correct number of communities. As purity of detected communities is likely to be higher when detected communities are small, we can still say that CND-autosim most correctly detected communities on every data sets.

5.3 Performance Comparison

To test the effectiveness of our proposed community detection method, we applied our method on real-world data sets. We used a combination of Jaccard’s index and affinity propagation as a method to be compared. Combination of raw data (edge weights as a similarity measure) and affinity propagation was compared as a baseline.

We also performed experiment using Newman’s leading eigenvector method. Newman’s leading eigenvector method is one of the most successful community detection method. This methods finds a partitioning that maximizes modularity of a partitioned graph [7]. The algorithm recursively divides the graph while the division increases the modularity of overall graph.

Another state-of-art method compared to our method was influence-based modularity devised by R. Ghosh et al. [4], which is an generalization of modularity based on measure of influence of a node to another nodes. The influence matrix is calculated as $\mathbf{A}(\mathbf{I} - \alpha\mathbf{A})^{-1}$, where \mathbf{A} is an affinity matrix.

We used College football [8], Political books [1], and Political blogs [11] data that are widely used to evaluate community detection methods. Parameters γ for CND and α for Ghosh’s method were chosen to a value that gives the highest modularity.

CND showed highest purity on College football data, and second, third highest purity on Political blogs and Political books data. CND showed competitive results compared to other methods, but the superiority of the method might seem not significant enough in some sense (Table 2).

However, the situation changes when we also consider the number of communities detected by community detection methods. Although Jaccard’s index combined with affinity propagation showed high purity, the method failed to detect correct number of communities from data sets. The rest of methods

¹ <http://www.orgnet.com/>

Table 2. Purity of communities detected by affinity propagation and Newman’s leading eigenvector method

Data Set	CND	Jaccard’s Index	Edge Weight	Newman’s	Ghosh’s
College Football	0.8345	0.8340	0.4739	0.5865	0.5808
Political Blogs	0.8357	0.8664	0.7776	0.8920	0.6457
Political Books	0.7548	0.7892	0.7471	0.5867	0.6351

Table 3. Number of communities detected by affinity propagation and Newman’s leading eigenvector method

Data Set (# communities)	CND	Jaccard’s Index	Edge Weight	Newman’s	Ghosh’s
College Football (12)	11	16	14	12	6
Political Blogs (2)	2	118	61	2	4
Political Books (3)	2	17	12	6	3

except R. Ghosh’s method also failed to detect correct number of communities on some of data sets (Table 3). In conclusion, CND combined with affinity propagation was the only method that finds correct number of communities with high accuracy from every data set tested.

6 Conclusion

We have presented a new similarity measure for social network analysis, referred to as CND from a simple intuition on communities in social networks. CND was devised to be used as an indicator for a pair of nodes whether they belong to a same community or not, using density of subgraph formed by their common neighbors. In addition, CND of a node and itself was interpreted as a measure of the likelihood of the node being an exemplar of a community. It gave a reasonable heuristic of choosing parameters for affinity propagation algorithm. Community detection using CND combined with affinity propagation showed higher purity of detected communities in multiple real-world data sets than other methods including Newman’s leading eigenvector method. The algorithm also detected the number of communities of data in a fair accuracy.

Acknowledgments

This work was supported by Korea Ministry of Knowledge Economy under the ITRC support program supervised by the National IT Industry Promotion Agency (NIPA-2009-C1090-0902-0045), ICRC for Artificial Neurosensory Device and Cognitive System, and KOSEF WCU Program (Project No. R31-2008-000-10100-0).

References

1. Adamic, L.A., Glance, N.: The political blogosphere and the 2004 U.S. election: Divided they blog. In: Proceedings of the 3rd International Workshop on Link Discovery, Chicago, Illinois (2005)
2. Fortunato, S., Castellano, C.: Community structure in graphs. In: Encyclopedia of Complexity and System Science. Springer, Heidelberg (2009)
3. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. *Science* 315 (February 2007)
4. Ghosh, R., Lerman, K.: Community detection using a measure of global influence. In: 2008 KDD workshop on Social Network Analysis, Las Vegas, Nevada, USA (2008)
5. Gustafsson, M., Hörnquist, M., Lombardi, A.: Comparison and validation of community structures in complex networks. *Physica A* 367, 559–576 (2006)
6. Klein, D.J., Randić, M.: Resistance distance. *Journal of Mathematical Chemistry* (1993)
7. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. *Physical Review E* 74 (2006)
8. Newman, M.E.J.: Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103(23), 8577–8582 (2006)
9. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Physical Review E* 69 (2004)
10. Wang, Y., Song, H., Wang, W., An, M.: A microscopic view on community detection in complex networks. In: Proceeding of the 2nd PhD Workshop on Information and Knowledge Management, Napa Valley, California (2008)

Divergence, Optimization and Geometry

Shun-ichi Amari

RIKEN Brain Science Institute, Hirosawa 2-1, Wako-shi, Saitama 351-0198, Japan
amari@brain.riken.jp

Abstract. Measures of divergence are used in many engineering problems such as statistics, mathematical programming, computational vision, and neural networks. The Kullback-Leibler divergence is its typical example which is defined between two probability distributions, and is invariant under information transformations. The Bregman divergence is another type of divergence, which are used often in optimization and signal processing. This is a class of divergences having dually flat geometrical structure. Divergence is often used for minimizing discrepancy between observed evidences and an underlying model. Projection to the model subspace plays a fundamental role. Here, geometry is important and dually flat geodesic structure is useful, because a generalized Pythagorean theorem and projection theorem hold.

1 Introduction

A divergence function is used in many engineering problems to show discrepancies between two objects and then to minimize a cost function having various constraints [1,10]. When probability distributions are discussed, the Kullback-Leibler (KL) divergence is a typical example of divergence, but there are many others divergences. For vision analysis, a picture is a two-dimensional array having non-negative components. Hence a divergence function of elements between two positive arrays will play an important role. We also treat positive-definite matrices, and a divergence function between two such matrices is used. In linear programming problems or more generally convex programming, a divergence between elements of a positive cone is important.

Let us show a simple problem: Given an element p from observed evidences, we search for an element q that is closest to p , among those satisfying constraints. Let M be a submanifold consisting of elements satisfying constraints. Then the candidate we search for is the minimizer of $D[p : q]$ under the constraint of $q \in M$, where $D[p : q]$ is a divergence between p and q .

When the entire space is Euclidean and the divergence is Euclidean distance, the optimal q is given by the projection of p to M . However, when geometry of the underlying space is not Euclidean and the divergence function is not Euclidean distance, we need a new geometrical framework.

A divergence provides a geometrical structure to the underlying manifold of engineering problems. We search for its differential geometrical background [5,3]. There are two types of divergence: One is invariant under information

transformation and the other is dually flat in the sense of geometry. Bregman divergence is of this latter type [6,7].

When the underlying space is dually flat, a generalized Pythagorean theorem and projection theorem hold. They provide useful tools for solving many engineering problems in computational vision, machine learning, mathematical programming, neural networks and others. See, for example, [6], [10], [19], [22], [24].

2 Divergence Function

A function $D[\mathbf{z} : \mathbf{y}]$ is called a divergence function defined in a space S , where $\mathbf{z}, \mathbf{y} \in S$ are two points in S , when it satisfies the following conditions:

- 1) $D[\mathbf{z} : \mathbf{y}] \geq 0$.
- 2) $D[\mathbf{z} : \mathbf{y}] = 0$, when and only when $\mathbf{z} = \mathbf{y}$.
- 3) For small $d\mathbf{z}$, Taylor expansion gives

$$D[\mathbf{z} + d\mathbf{z} : \mathbf{z}] \approx \frac{1}{2} \sum g_{ij} dz_i dz_j, \tag{1}$$

where $(g_{ij}(\mathbf{z}))$ is a positive-definite matrix.

A divergence is not necessarily symmetric with respect to \mathbf{z} and \mathbf{y} , and it does not satisfy the triangular inequality. Hence, it is not a distance.

The square of the Euclidean distance

$$D[\mathbf{z} : \mathbf{y}] = \frac{1}{2} \sum |z_i - y_i|^2 \tag{2}$$

is a trivial example of divergence. The Kullback-Leibler divergence

$$KL[\mathbf{p} : \mathbf{q}] = \sum_{i=0}^n p_i \log \frac{p_i}{q_i}, \tag{3}$$

defined in the space S_n of probability distributions, $\mathbf{p} = (p_0, p_1, \dots, p_n)$, $\sum p_i = 1$, is another example.

3 Invariant Divergences

The invariance principle for defining geometry of S_n was proposed by Chentsov [9]. It is further developed in information geometry (Amari and Nagaoka, [5]). Here, we show its version used by Csiszár [12][13][11].

3.1 Information Monotonicity

For a probability distribution $\mathbf{p} = (p_0, p_1, \dots, p_n)$ over $X = \{x_0, x_1, \dots, x_n\}$, we divide X into m subsets, $G_1, G_2, \dots, G_m (m < n + 1)$. It is a partition of X ,

$$X = \cup G_i, \tag{4}$$

$$G_i \cap G_j = \phi. \tag{5}$$

Assume that we do not know x_i but know which subset G_j it belongs to. This is coarse-graining of X .

The coarse-graining generates a new probability distributions $\bar{\mathbf{p}} = (\bar{p}_1, \dots, \bar{p}_m)$ over G_1, \dots, G_m ,

$$\bar{p}_j = \text{Prob}\{G_j\} = \sum_{x_i \in G_j} \text{Prob}\{x_i\}. \tag{6}$$

Let $\bar{D}[\bar{\mathbf{p}} : \bar{\mathbf{q}}]$ be an induced divergence between $\bar{\mathbf{p}}$ and $\bar{\mathbf{q}}$. Since detailed information is lost by coarse-graining, it is natural to assume

$$\bar{D}[\bar{\mathbf{p}} : \bar{\mathbf{q}}] \leq D[\mathbf{p} : \mathbf{q}]. \tag{7}$$

For two distributions \mathbf{p} and \mathbf{q} , assume that the outcome x_i is known to belong to G_j . How much information is obtained to distinguish the two probability distributions \mathbf{p} and \mathbf{q} by knowing further detail? Since x_i is known to belong to subset G_j , we consider the conditional probability distributions

$$p(x_i | G_j), \quad q(x_i | G_j). \tag{8}$$

If they are equal, we cannot obtain further information to distinguish \mathbf{p} from \mathbf{q} by observing the outcome x_i inside G_j . Hence,

$$\bar{D}[\bar{\mathbf{p}} : \bar{\mathbf{q}}] = D[\mathbf{p} : \mathbf{q}] \tag{9}$$

holds, when and only when

$$p(x_i | G_j) = q(x_i | G_j). \tag{10}$$

A divergence satisfying the above requirements is called an invariant divergence, and such a property is termed as information monotonicity.

3.2 f -Divergence and Information Monotonicity

When a divergence is written as a sum of functions of two variables p_i and q_i : $D[\mathbf{p} : \mathbf{q}] = \sum_{i=0}^n D(p_i, q_i)$, it is called a separable divergence.

Given a convex function $f(u)$, f -divergence is defined by

$$D_f[\mathbf{p} : \mathbf{q}] = \sum_{i=0}^n p_i f\left(\frac{q_i}{p_i}\right), \tag{11}$$

where we assume that $f(1) = 0, f'(1) = 0, f''(1) = 1$ [11,23]. This is a separable divergence.

Csiszár [12,13] found that an f -divergence satisfies information monotonicity. Moreover, the class of f -divergences is unique in the sense that any separable divergence satisfying the information monotonicity is an f -divergence.

Theorem 1. An f -divergence satisfies the information monotonicity. Conversely, any separable information monotonic divergence is written in the form of f -divergence.

The proof is found, e.g., in Amari [4].

3.3 α -Divergence in S_n

The α -divergence is a special case of f -divergence, defined by the following f -function.

$$f_\alpha(u) = \frac{4}{1-\alpha^2} \left(1 - u^{\frac{1+\alpha}{2}}\right) - \frac{2}{1-\alpha}(u-1). \tag{12}$$

It is given by

$$D_\alpha[\mathbf{p} : \mathbf{q}] = \frac{4}{1-\alpha^2} \left(1 - \sum p_i^{\frac{1+\alpha}{2}} q_i^{\frac{1-\alpha}{2}}\right). \tag{13}$$

The α -divergence was introduced by Havdra and Charvát [17], and has been studied extensively by Amari and Nagaoka [5]. Its applications were described earlier in Chernoff [8], and later in Matsuyama [18], Amari [2], etc. to mention a few. It is the squared Hellinger distance for $\alpha = 0$, and the KL-divergence and its reciprocal are obtained in the limit of $\alpha \rightarrow \pm 1$,

$$KL[\mathbf{p} : \mathbf{q}] = \sum p_i \log \frac{q_i}{p_i}. \tag{14}$$

3.4 α -Divergence in M_{n+1}

Let M_{n+1} is a space of positive measures, $\mathbf{m} = (m_0, m_1, \dots, m_n)$, where $m_i > 0$ and we do not require $\sum m_i = 1$. S_n is its subspace satisfying the condition $\sum m_i = 1$. Then, an f -divergence is defined similarly in M_{n+1} [5], and the α -divergence is

$$D_\alpha[\mathbf{m} : \mathbf{n}] = \frac{4}{1-\alpha^2} \sum \left(\frac{1-\alpha}{2} m_i + \frac{1+\alpha}{2} n_i - m_i^{\frac{1+\alpha}{2}} n_i^{\frac{1-\alpha}{2}} \right). \tag{15}$$

4 Dually Flat Divergences —Bregman Divergence

4.1 Bregman Divergence

Let $\varphi(\mathbf{z})$ be a strictly convex differentiable function. Then,

$$D_\varphi(\mathbf{z}, \mathbf{y}) = \varphi(\mathbf{z}) - \varphi(\mathbf{y}) - \nabla\varphi(\mathbf{y}) \cdot (\mathbf{z} - \mathbf{y}) \tag{16}$$

satisfies the conditions of a divergence, where $\nabla\varphi(\mathbf{y})$ is the gradient of φ . This is called the Bregman divergence [3,6,7].

In the case of probability distributions S_n , when we put

$$\varphi(\mathbf{z}) = \sum z_i \log z_i, \tag{17}$$

we then obtain the KL-divergence as the corresponding Bregman divergence. When $\varphi(\mathbf{z}) = (1/2) \sum z_i^2$, we have the squared Euclidean distance.

4.2 Manifold M_{n+1} of Positive Measures

Let us put

$$r_\alpha(u) = \begin{cases} \frac{2}{1-\alpha} \left(u^{\frac{1-\alpha}{2}} - 1 \right), & \alpha \neq 1, \\ \log u, & \alpha = 1. \end{cases} \tag{18}$$

Consider

$$z_i = r_\alpha(p_i) \tag{19}$$

and use $\mathbf{z} = (z_i)$ as a new coordinate system of M_{n+1} . We define

$$\varphi_\alpha(\mathbf{z}) = \frac{2}{1+\alpha} \sum_i r_\alpha^{-1}(z_i) \tag{20}$$

$$= \frac{2}{1+\alpha} \sum \left(1 + \frac{1-\alpha}{2} z_i \right)^{\frac{2}{1-\alpha}}, \tag{21}$$

$\alpha \neq \pm 1.$

Then, this is a convex function of \mathbf{z} . This generates the α -divergence in M_{n+1} .

Theorem 2. The α -divergence is unique in the same that it is an f -divergence and Bregman divergence at the same time.

4.3 Manifold of Positive-Definite Matrices

When $|P|$ is the determinant of P ,

$$\varphi(P) = -\log |P| \tag{22}$$

is a convex function of P in the set P_n of $n \times n$ positive-definite matrices. Its gradient is

$$\nabla \varphi(P) = -P^{-1}. \tag{23}$$

Hence, the induced divergence is

$$D[P : Q] = -\log |PQ^{-1}| + \text{tr}(PQ^{-1}) - n, \tag{24}$$

where the operator tr is the trace of a matrix.

Quantum information geometry (Amari and Nagaoka [5], Grasselli [15], Hasegawa [16], Petz [20]) uses the convex function

$$\varphi(P) = \text{tr}(P \log P - P). \tag{25}$$

Its gradient is

$$\nabla \varphi(P) = \log P. \tag{26}$$

The divergence is

$$D(P : Q) = \text{tr} \{ P (\log P - \log Q) + P - Q \}, \tag{27}$$

which is the von Neumann divergence.

We further define the following function by using a convex function f ,

$$\varphi_f(P) = \text{tr}f(P) = \sum f(\lambda_i), \tag{28}$$

where λ_i are the eigenvalues of P . Then, φ_f is a convex function of P , from which we derive a dual geometrical structure depending on f (Dhillon and Tropp [14]).

More generally, by putting

$$\varphi_\alpha(\lambda) = \frac{-4}{1-\alpha^2} \left(\lambda^{\frac{1+\alpha}{2}} - \lambda \right), \quad (-1 < \alpha < 1) \tag{29}$$

we have the α -divergence,

$$D_\alpha[P : Q] = \frac{4}{1-\alpha^2} \text{tr} \left[\frac{1-\alpha}{2} P + \frac{1+\alpha}{2} Q - P^{\frac{1+\alpha}{2}} Q^{\frac{1-\alpha}{2}} \right] \tag{30}$$

(Hasegawa, [16]). This is a generalization of the α -divergence defined in the space of positive measures.

4.4 Dual Structure Derived from Bregman Divergence

We search for a pair of dual affine coordinate systems with a Bregman divergence, by using the Legendre transformation. Given a convex function $\varphi(\mathbf{z})$, we consider \mathbf{z} as an affine coordinate system, so that a geodesic $\mathbf{z}(t)$ is of the form $\mathbf{z}(t) = t\mathbf{a} + \mathbf{b}$. Let us define \mathbf{z}^* by

$$\mathbf{z}^* = \nabla\varphi(\mathbf{z}). \tag{31}$$

We can then define the dual function of φ by

$$\varphi^*(\mathbf{z}^*) = \max_{\mathbf{z}} \{ \mathbf{z} \cdot \mathbf{z}^* - \varphi(\mathbf{z}) \}, \tag{32}$$

which is a convex function of \mathbf{z}^* . We can describe the geometry of S by using the dual convex function φ^* and the dual coordinates \mathbf{z}^* . The coordinate system \mathbf{z}^* is considered as a dual affine coordinate system. Obviously, \mathbf{z} and \mathbf{z}^* are dual, since we have

$$\mathbf{z} = \nabla\varphi^*(\mathbf{z}^*). \tag{33}$$

The dual function $\varphi^*(\mathbf{z}^*)$ induces a divergence,

$$D^*[\mathbf{y}^* : \mathbf{z}^*] = \varphi^*(\mathbf{y}^*) - \varphi^*(\mathbf{z}^*) - \nabla\varphi^*(\mathbf{z}^*) \cdot (\mathbf{y}^* - \mathbf{z}^*). \tag{34}$$

Theorem 3. The two divergences D and D^* are mutually reciprocal in the sense

$$D^*[\mathbf{y}^* : \mathbf{z}^*] = D[\mathbf{z} : \mathbf{y}]. \tag{35}$$

This shows that the two divergences are essentially the same, and \mathbf{z}^* is the dual affine coordinate system. The divergence is written in the dual form

$$D[\mathbf{z} : \mathbf{y}] = \varphi(\mathbf{z}) + \varphi^*(\mathbf{y}^*) - \mathbf{z} \cdot \mathbf{y}^*. \tag{36}$$

5 Optimization and Projection

A divergence function is used in many applications. One is to define the center of points z_1, \dots, z_k which form a cluster. The center is defined by

$$\bar{z} = \arg \min_z \{D(z, z_1), \dots, D(z, z_k)\} \tag{37}$$

or

$$\bar{z} = \arg \min_z \sum_{i=1}^k D(z, z_i). \tag{38}$$

Another problem is the following: Given a point z_0 , we search for the point \bar{z} that minimizes the divergence $D[z_0 : z]$, where z belongs to a submanifold M ,

$$\bar{z} = \arg \min_{z \in M} D[z_0 : z]. \tag{39}$$

This is solved by the projection of z_0 to M in a dully flat manifold.

To solve this problem, we show the following theorem.

Pythagorean Theorem [5]. Let P, Q, R be three points in a dually flat manifold S whose coordinates (and dual coordinates) are represented by z_P, z_Q, z_R (z_P^*, z_Q^*, z_R^*), respectively. When the dual geodesic connecting P and Q is orthogonal at Q to the geodesic connecting Q and R , then

$$D[P : R] = D[P : Q] + D[Q : R]. \tag{40}$$

Dually, when the geodesic connecting P and Q is orthogonal at Q to the dual geodesic connecting Q and R , we have

$$D[R : P] = D[Q : P] + D[R : Q]. \tag{41}$$

Proof. By using (36), we have

$$\begin{aligned} & D[R : Q] + D[Q : P] \\ &= \varphi(z_R) + \varphi^*(z_Q^*) + \varphi(z_Q) + \varphi^*(z_P^*) \\ &\quad - z_R \cdot z_Q^* - z_Q \cdot z_P^* \end{aligned} \tag{42}$$

$$\begin{aligned} &= \varphi(z_R) + \varphi^*(z_P^*) + z_Q \cdot z_Q^* - z_R \cdot z_Q^* \\ &\quad - z_Q \cdot z_P^* \end{aligned} \tag{43}$$

$$= D[z_R : z_P^*] + (z_Q - z_R) \cdot (z_Q^* - z_P^*). \tag{44}$$

The tangent vector of the geodesic connecting Q and R is $z_Q - z_R$, and the tangent vector of the dual geodesic connecting Q and P is $z_Q^* - z_P^*$ in the dual coordinate system. Hence, the second term of the right-hand side of the above equation vanishes, because the primal and dual geodesics connecting Q and R , and Q and P are orthogonal.

The following projection theorem [5] is a consequence of the generalized Pythagorean theorem. Let M be a smooth submanifold of S . Given a point P

outside M , we connect it to a point Q in M by geodesic (dual geodesic). When the geodesic (dual geodesic) connecting P and Q is orthogonal to M (that is, orthogonal to any tangent vectors of M), Q is said to be the geodesic projection (dual geodesic projection) of P to M .

Projection Theorem. Given P and M , the point Q (Q^*) that minimizes divergence $D(P : R)$, $R \in Q$ ($D(R : P)$, $R \in M$), is the projection (dual projection) of P to Q .

This theorem is useful, when we search for the point belonging to M that minimizes the divergence $D(P : Q)$ or $D(Q : P)$ for preassigned P . In many engineering problems, P is given from observed data, and M is a model to describe the underlying structure.

6 Conclusion

We have studied various types of a divergence functions. The f -divergences are unique information-monotone divergences, which gives the α -structure of information geometry. On the other hand Bregman divergences are characterized by the dually flat geometrical structure. The Kullback-Leibler divergence is the unique intersection of classes of f -divergences and Bregman divergences in the manifold of probability distributions. However, the α -divergences are unique intersection of these classes in the manifold of positive measure. We have also shown applications of divergences, in particular the projection structure.

References

1. Ali, M., Silvey, S.: A general class of coefficients of divergence of one distribution from another. *Journal of Royal Statistical Society Ser B*(28), 131–142 (1966)
2. Amari, S.: Integration of stochastic models by minimizing α -divergence. *Neural Computation* 19, 2780–2796 (2007)
3. Amari, S.: Information geometry and its applications: Convex function and dually flat manifold. In: Nielsen, F. (ed.) *Emerging Trends in Visual Computing*. LNCS, vol. 5416, pp. 75–102. Springer, Heidelberg (2009)
4. Amari, S.: Alpha divergence is unique, belonging to both classes of f -divergence and Bregman divergence. *IEEE Trans. on Information Theory* 55 (November 2009)
5. Amari, S., Nagaoka, H.: *Methods of Information Geometry*. Oxford University Press, New York (2000)
6. Banerjee, S., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with Bregman divergences. *J. Machine Learning Research* 6, 1705–1749 (2005)
7. Bregman, L.: The relaxation method of finding a common point of convex sets and its application to the solution of problems in convex programming. *Comp. Math. Phys., USSR*. 7, 200–217 (1967)
8. Chernoff, H.: A measure of asymptotic efficiency for tests of a hypothesis based on a sum of observations. *Annals of Mathematical Statistics* 23, 493–507 (1952)
9. Chentsov, N.N.: *Statistical Decision Rules and Optimal Inference*, Rhode Island, U.S.A. ; originally published in Russian, Nauka, Moscow, 1972. American Mathematical Society (1982)

10. Cichocki, A., Adunek, R., Phan, A.H., Amari, S.: Nonnegative Matrix and Tensor Factorizations. John Wiley, Chichester (2009)
11. Csiszár, I.: Information-type measures of difference of probability distributions and indirect observations. *Studia Sci. Math. Hungar.* 2, 299–318 (1967)
12. Csiszár, I.: Why least squares and maximum entropy? An axiomatic approach to inference for linear problems. *Annals of Statistics* 19, 2032–2066 (1991)
13. Csiszár, I.: Axiomatic characterizations of information measures. *Entropy* 10, 261–273 (2008)
14. Dhillon, I.S., Tropp, J.A.: Matrix nearness problem with Bregman divergences. *SIAM J. on Matrix Analysis and Applications*. 29, 1120–1146 (2007)
15. Grasselli, M.R.: Duality, monotonicity and Wigner-Yanase-Dyson metrics. *Infinite Dimensional Analysis, Quantum Probability and Related Topics* 7, 215–232 (2004)
16. Hasegawa, H.: α -divergence of the non-commutative information geometry. *Reports on Mathematical Physics* 33, 87–93 (1993)
17. Havrda, J., Charvát, F.: Quantification method of classification process. Concept of structural α -entropy. *Kybernetika*. 3, 30–35 (1967)
18. Matsuyama, Y.: The α -EM algorithm: Surrogate likelihood maximization using α -logarithmic information measures. *IEEE Trans. on Information Theory* 49, 672–706 (2002)
19. Nielsen, F. (ed.): *Emerging Trends in Visual Computing*. LNCS, vol. 5416. Springer, Heidelberg (2009)
20. Petz, D.: Monotone metrics on matrix spaces. *Linear Algebra and its Applications* 244, 81–96 (1996)
21. Rényi, A.: On measures of entropy and information. In: *Proc. 4th Berk. Symp. Math. Statist. and Probl.*, vol. 1, pp. 547–561. University of California Press, Berkeley (1961)
22. Si, S., Tao, D., Geng, B.: Bregman divergence based regularization for transfer subspace learning. *IEEE Transactions on Knowledge and Data Engineering*
23. Taneja, I., Kumar, P.: Relative information of type s , Csiszár's f -divergence, and information inequalities. *Information Sciences* 166, 105–125 (2004)
24. Tao, D., Li, X., Wu, X., Maybank, S.J.: Geometric mean for subspace selection. *IEEE Transactions on Pattern Analysis and Machine intelligence* 31(2), 260–274 (2009)

Robust Stability of Fuzzy Cohen-Grossberg Neural Networks with Delays

Tingwen Huang¹ and Zhigang Zeng²

¹ Texas A and M University at Qatar, Doha, P.O. Box 5825, Qatar
tingwen.huang@qatar.tamu.edu

² Department of Control Science and Engineering,
Huazhong University of Science and Technology,
Wuhan, Hubei, 430074, China
zgzen@mail.hust.edu.cn

Abstract. In the paper, a new exponentially robust stability criterion for interval fuzzy Cohen-Grossberg type neural networks with time-varying delays is obtained by using Lyapunov-Krasovskii functional with the differential inequality and linear matrix inequality technique. The new criterion is easily verifiable.

Keywords: Fuzzy; Cohen-Grossberg neural networks; Delays; Linear matrix inequality.

1 Introduction

In the past decades, the dynamics of Cohen-Grossberg neural networks (CGNN), a generalized type of cellular neural networks [1-26] without delay or with delays have been extensively studied due to their promising potential applications in classification and parallel computing. The qualitative analysis of these dynamical behaviors is important to the practical design and applications of neural networks since such applications depend on the existence and uniqueness of equilibrium points and the qualitative properties of stability.

At the same time, T. Yang et al. [21-23] introduced fuzzy neural networks which combine the fuzzy logic with the traditional neural networks. Fuzzy neural networks could be used in image processing and pattern recognition. In practice, the stability of fuzzy neural networks is very important as that of traditional neural networks. T. Yang et al. [21-23] have investigated the existence and uniqueness of the equilibrium point and the stability of fuzzy neural networks without any delays. Realistic modeling of many large neural networks with non-local interaction inevitably have connection delays which naturally arise as a consequence of finite information transmission and processing speeds among the neurons. Thus, it is natural to consider delayed neural networks. Instability of the delayed neural networks could be caused by time-delays, so lots of deep investigations have been done on the stability of delayed neural networks. Liu et al. [17] have investigated fuzzy neural networks with time-varying delays and Huang et al. [10] have investigated the stability of fuzzy neural networks with

diffusion term. Song et al. [19] introduced and investigated the impulsive effects on the stability of fuzzy Cohen-Grossberg neural networks with time-delays.

The stability of CGNN could become unstable because of the existence of the unavoidable modeling errors, external disturbance and parameter fluctuation during the implementation on very large scale integration chips. This implies that a nice neural network should have certain robustness to against such errors, disturbance and fluctuation. To deal this problem faced by neural networks with uncertainty, Liao et al. [15] introduced interval neural networks. After that, some researchers obtained some criteria for robust stability of neural networks with delays or without delays. Recently, Liao et al [14] and Li et al. [12] further investigated interval neural networks with time delays using Lyapunov-Krasovskii functional with the differential inequality and linear matrix inequality techniques. However, to the best of our knowledge, no result on robust stability of interval fuzzy Cohen-Grossberg type neural networks has been reported in the literature so far. In this paper, we would present some sufficient conditions to guarantee the interval fuzzy neural networks being robustly stable.

The rest of the paper is as follows. In Section 2, problem formulation and preliminaries are given. In Section 3, several sufficient criteria are obtained. Finally, conclusions are drawn in Section 4.

2 Problem Formulation and Preliminaries

In this paper, we would like to consider fuzzy neural networks with time-varying delay described by the following form:

$$\begin{aligned} \frac{dx_i}{dt} = & a_i(x_i(t))[-c_i(x_i(t)) + \sum_{j=1}^n \xi_{ij} f_j(x_j(t)) + \bigwedge_{j=1}^n \gamma_{ij} f_j(x_j(t - \tau_j(t)))] \\ & + \sum_{j=1}^n b_{ij} \mu_j + G_i + \bigwedge_{j=1}^n T_{ij} \mu_j \\ & + \bigvee_{j=1}^n \delta_{ij} f_j(x_j(t - \tau_j(t))) + \bigvee_{j=1}^n H_{ij} \mu_j], \quad i = 1, \dots, n, \end{aligned} \tag{1}$$

where $a_i(x_i)$ represents the amplification function; $\gamma_{ij}, \delta_{ij}, T_{ij}$ and H_{ij} are elements of fuzzy feedback MIN template, fuzzy feedback MAX template, fuzzy feed forward MIN template and fuzzy feed forward MAX template, respectively; b_{ij} are elements of feed forward template; \bigwedge and \bigvee denote the fuzzy AND and fuzzy OR operation, respectively; x_i, μ_i and G_i denote state, input and bias of the i th neuron, respectively; f_i is the activation function; $\tau_i(t) \geq 0$ is the transmission delay vector with $\tau_i(t) \leq \tau$ where τ is a positive constant.

For the above model (1), we can write it as the following matrix-vector form:

$$\begin{aligned} \frac{dx(t)}{dt} = & \bar{A}(x(t))[-\bar{C}(x(t)) + \Xi f(x(t)) + \Gamma \bar{\bigwedge} f(x(t - \tau(t)))] + B\mu + G \\ & + T \bar{\bigwedge} \mu + \Delta \bar{\bigvee} f(x(t - \tau(t))) + H \bar{\bigvee} \mu] \end{aligned} \tag{2}$$

where $x(t) = (x_1(t), x_2(t), \dots, x_n(t))^T$, $\mu(t) = (\mu_1(t), \mu_2(t), \dots, \mu_n(t))^T$, $\overline{A}x(t) = \text{diag}(a_1(x_1(t)), a_2(x_2(t)), \dots, a_n(x_n(t)))$, $\overline{C}x(t) = \text{diag}(c_1(x_1(t)), \dots, c_n(x_n(t)))$, $\Xi = (\xi_{ij})_{n \times n}$, $B = (b_{ij})_{n \times n}$, $G = \text{diag}(G_1, \dots, G_n)$, $\Gamma = (\gamma_{ij})_{n \times n}$, $\Delta = (\delta_{ij})_{n \times n}$, $T = (T_{ij})_{n \times n}$, $H = (H_{ij})_{n \times n}$, $\overline{\wedge}$ is a diagonal matrix with diagonal elements: fuzzy AND operation \wedge , $\overline{\vee}$ is a diagonal matrix with diagonal elements: fuzzy OR operation \vee .

In this paper, we assume the following:

(H₁): the activation function, $f = (f_1, \dots, f_n)^T$, f_i are bounded functions defined on R and satisfy

$$|f_i(x) - f_i(y)| \leq l_i|x - y|, \quad i = 1, \dots, n, \tag{3}$$

for any $x, y \in R$, where l_i are positive constants.

(H₂): $a_i(\cdot)$, $i = 1, \dots, n$, are continuous and there exist positive constants m and M , such that $m \leq a_i(x) \leq M$ for all $x \in R$.

(H₃): $c'_i(x) \geq d_i > 0$ for all $x \in R$.

For convenience, we use $A^T, A^{-1}, \lambda_m(A), \lambda_M(A)$ to denote the transpose of, inverse of, the minimum eigenvalue, the maximum eigenvalue of a square matrix A , respectively. The vector norm is taken to be Euclidian, denoted by $\|\cdot\|$. And $A > 0 (< 0, \leq 0, \geq 0)$ denotes symmetrical positive (negative, semi-negative, semi-positive) definite matrix A . Moreover, if $A = (a_{ij})_{n \times n}$, then $|A| = (|a_{ij}|)_{n \times n}$.

In the practical implementation of neural networks, in general, the deviations and perturbations of the weights of the connections are bounded. Therefore, the quantities of the coefficients $d_i, \xi_{ij}, \gamma_{ij}$ and δ_{ij} may be intervalized as follow:

$$\begin{aligned} D_I &:= [\underline{D}, \overline{D}] = \{D = \text{diag}(d_i) : \underline{D} \leq D \leq \overline{D}, \text{i.e., } \underline{d}_i \leq d_i \leq \overline{d}_i, i = 1, \dots, n\} \\ \Xi_I &:= [\underline{\Xi}, \overline{\Xi}] = \{\Xi = (\xi_{ij})_{n \times n} : \underline{\Xi} \leq \Xi \leq \overline{\Xi}, \text{i.e., } \underline{\xi}_{ij} \leq \xi_{ij} \leq \overline{\xi}_{ij}, i = 1, \dots, n\} \\ \Gamma_I &:= [\underline{\Gamma}, \overline{\Gamma}] = \{\Gamma = (\gamma_{ij})_{n \times n} : \underline{\Gamma} \leq \Gamma \leq \overline{\Gamma}, \text{i.e., } \underline{\gamma}_{ij} \leq \gamma_{ij} \leq \overline{\gamma}_{ij}, i = 1, \dots, n\} \\ \Delta_I &:= [\underline{\Delta}, \overline{\Delta}] = \{\Delta = (\delta_{ij})_{n \times n} : \underline{\Delta} \leq \Delta \leq \overline{\Delta}, \text{i.e., } \underline{\delta}_{ij} \leq \delta_{ij} \leq \overline{\delta}_{ij}, i = 1, \dots, n\} \end{aligned} \tag{4}$$

Moreover, for convenience, we define, for $i, j = 1, \dots, n$,

$$\begin{aligned} \xi_{ij}^* &= \max\{|\underline{\xi}_{ij}|, |\overline{\xi}_{ij}|\}, \omega_{ij}^* = \max\{|\underline{\gamma}_{ij}| + |\underline{\delta}_{ij}|, |\overline{\gamma}_{ij}| + |\overline{\delta}_{ij}|\} \\ \xi_i^* &= \sum_{j=1}^n (\xi_{ij}^* \sum_{k=1}^n \xi_{kj}^*), \omega_i^* = \sum_{j=1}^n (\omega_{ij}^* \sum_{k=1}^n \omega_{kj}^*), \\ \Xi^* &= \text{diag}(\xi_i^*), \Omega^* = \text{diag}(\omega_i^*). \end{aligned} \tag{5}$$

It is noted that bounded activation functions always guarantee the existence of an equilibrium point for model (1). Let $x^* = (x_1^*, \dots, x_n^*)^T$ be an equilibrium point of model (1) for a given μ . To simplify the proof, we shift the equilibrium point x^* of (1) to the origin by using the transformation $y(t) = x(t) - x^*$. Model (1) becomes the following form:

$$\begin{aligned} \frac{dy_i}{dt} = & a_i(y_i(t) + x_i^*)[-(c_i(y_i(t) + x_i^*) - c_i(x_i^*)) + \sum_{j=1}^n \xi_{ij}(f_j(x_j(t)) - f_j(x_j^*)) \\ & + \bigwedge_{j=1}^n \gamma_{ij} f_j(x_j(t - \tau_j(t))) - \bigwedge_{j=1}^n \gamma_{ij} f_j(x_j^*) \\ & + \bigvee_{j=1}^n \delta_{ij} f_j(x_j(t - \tau_j(t))) - \bigvee_{j=1}^n \delta_{ij} f_j(x_j^*)], \quad i = 1, \dots, n \end{aligned} \tag{6}$$

or matrix-vector form:

$$\begin{aligned} \frac{dy(t)}{dt} = & \bar{A}(y(t) + x^*)[-(\bar{C}(y(t) + x^*) - \bar{C}x^*) + \Xi(f(x(t)) - f(x^*)) + \\ & \Gamma \bar{\bigwedge} f(x(t - \tau(t))) - \Gamma \bar{\bigwedge} f(x^*) + \Delta \bar{\bigvee} f(x(t - \tau(t))) - \Delta \bar{\bigvee} f(x^*)] \end{aligned} \tag{7}$$

Definition 1. The equilibrium point x^* of (1) is said to be globally exponentially stable if there exist constants $\lambda > 0$ and $K > 0$ such that

$$\|u_i(t) - x_i^*\| \leq K \max_{1 \leq i \leq n} \|\varphi_i - x_i^*\| e^{-\lambda t} \tag{8}$$

for all $t \geq 0$, where $\|\varphi_i - x_i^*\| = \sup_{s \in (-\tau, 0]} |\varphi_i(s) - x_i^*|$, $i = 1, \dots, n$.

Definition 2. Model (1) is said to be robustly exponentially stable if its unique equilibrium point $u^* \in R^n$ is globally exponentially stable for any $D \in D_I$, $\Gamma \in \Gamma_I$, $\Delta \in \Delta_I$.

Definition 3. For any continuous function $f : R \rightarrow R$, its Dini’s time-derivative is defined as

$$\dot{f}(t) = \limsup_{\Delta t \rightarrow 0} \frac{f(t + \Delta t) - f(t)}{\Delta t}$$

In order to get the main result regarding the robustly exponential stability of model (1), we would like to present several lemmas first.

Lemma 1. ([21]). For any $a_{ij} \in R$, $x_j, y_j \in R$, $i, j = 1, \dots, n$, we have the following estimations,

$$\left| \bigwedge_{j=1}^n a_{ij} x_j - \bigwedge_{j=1}^n a_{ij} y_j \right| \leq \sum_{1 \leq j \leq n} (|a_{ij}| \cdot |x_j - y_j|) \tag{9}$$

and

$$\left| \bigvee_{j=1}^n a_{ij} x_j - \bigvee_{j=1}^n a_{ij} y_j \right| \leq \sum_{1 \leq j \leq n} (|a_{ij}| \cdot |x_j - y_j|) \tag{10}$$

Lemma 2. ([13]). Given any real matrices A, B, C with appropriate dimensions and C is a positive symmetric matrix. Then, for any scalar $\varepsilon > 0$, the following inequality holds:

$$A^T B + B^T A \leq \varepsilon A^T C A + \frac{1}{\varepsilon} B^T C^{-1} B$$

Lemma 3. (Schur complement, Boyd et al. [27]). The following LMI:

$$0 < \begin{bmatrix} A(x) & B(x) \\ B^T(x) & C(x) \end{bmatrix}$$

where $A(x) = A^T(x)$, $C(x) = C^T(x)$, and $B(x)$ depend affinely on x , is equivalent to one of the following conditions:

- (i) $A(x) > 0$, $C(x) - B^T(x)A(x)^{-1}B(x) > 0$;
- (i) $C(x) > 0$, $A(x) - B(x)C(x)^{-1}B^T(x) > 0$;

Lemma 4. For any $\Xi = (\xi_{ij})_{n \times n} \in \Xi_I$, $\Gamma = (\gamma_{ij})_{n \times n} \in \Gamma_I$, $\Delta = (\delta_{ij})_{n \times n} \in \Delta_I$, we have $\Xi \Xi^T \leq \Xi^*$, $(|\Gamma| + |\Delta|)(|\Gamma| + |\Delta|)^T \leq \Omega^*$, where Ξ^* , Ω^* are defined by (5).

It is clear that Ξ^* , Ω^* are positive matrix and $\Xi^* - \xi \xi^T$, $\Omega^* - (|\Gamma| + |\Delta|)(|\Gamma| + |\Delta|)^T$ diagonally dominant. Thus, $\Xi^* - \xi \xi^T$, $\Omega^* - (|\Gamma| + |\Delta|)(|\Gamma| + |\Delta|)^T$ are positive matrices, the above lemma results follow.

3 Robust Stability Criterion

In this section, we will use the Lyapunov method, LMI matrix inequality techniques to obtain the sufficient condition for the robust stability of the equilibrium point of fuzzy neural networks with time-varying delays. The main result is presented as the following theorem.

Theorem 1. Suppose that there exist positives: u, v, w and positive diagonal matrix P such that

- (i) The LMI holds:

$$0 < \begin{bmatrix} 2mP\underline{D} - wP - \frac{M}{u}\Xi^* - \frac{M}{v}\Omega^* & \sqrt{Mu}PL \\ \sqrt{Mu}LP & I_n \end{bmatrix}$$

- (ii) $c \equiv \frac{Mv}{w} \lambda_M(P) \max_{1 \leq i \leq n} \{l_i^2\} < 1$

where I_n is an n th-order identity matrix, $L = \text{diag}(l_i)_{n \times n}$ and l_i being the Lipschitz constants in (3), $\underline{D}, \Xi^*, \Omega^*$ defined in (4). Then, model (1) is exponentially robustly stable under the assumption H .

Proof: It is clear that the origin is an equilibrium point of system (6). We consider the following Lyapunov-Krasovskii functional:

$$V(y(t)) = y^T(t)Qy(t), \tag{11}$$

where $Q = P^{-1}$ and $y(t) = x(t) - x^*$.

It is obvious that

$$\lambda_m(Q) \|y(t)\|^2 \leq V(y(t)) \leq \lambda_M(Q) \|y(t)\|^2 \tag{12}$$

To simplify the notation, we define

$$a_\tau = \sup_{-\tau \leq t \leq \tau} \{\|y(t)\|\}, \quad a_{2\tau} = \sup_{-\tau \leq t \leq 2\tau} \{\|y(t)\|\}.$$

Since the solution $y(t)$ of system (6) is continuous, the existence of $a_\tau, a_{2\tau}$ is guaranteed. To obtain the inequalities of the Dini's time derivative of $V(y(t))$ along the trajectory of system (6), we need to use Lemma 1 to Lemma 4 and the assumptions $(H_1) - (H_3)$ in the following process:

$$\begin{aligned} \dot{V}(y(t)) &= 2y^T(t)Q\overline{A}(y(t) + x^*)[-(\overline{C}(y(t) + x^*) - \overline{C}x^*) + \Xi(f(x(t)) - f(x^*)) \\ &\quad + \Gamma\overline{\bigwedge}f(x(t - \tau(t))) - \Gamma\overline{\bigwedge}f(x^*) + \Delta\overline{\bigvee}f(x(t - \tau(t))) - \Delta\overline{\bigvee}f(x^*)] \\ &\leq -2my^T(t)QDy(t) + 2y^T(t)Q\overline{A}(y(t) + x^*)\Xi(f(x(t)) - f(x^*)) + \\ &\quad 2M|y^T(t)|Q|\Gamma||f(x(t)) - f(x^*)| + 2M|y^T(t)|Q|\Delta||f(x(t)) - f(x^*)| \\ &\leq -2my^T(t)Q\underline{D}y(t) + \frac{M}{u}y^T(t)Q\underline{\Xi}\Xi^TQy(t) + Mu(f(x(t)) - f(x^*))^T \\ &\quad \times (f(x(t)) - f(x^*)) + \frac{M}{v}y^T(t)Q(|\Gamma| + |\Delta|)(|\Gamma| + |\Delta|)^TQy(t) \\ &\quad + Mv(f(x(t - \tau(t))) - f(x^*))^T \times (f(x(t - \tau(t))) - f(x^*)) \\ &\leq -2my^T(t)Q\underline{D}y(t) + \frac{M}{u}y^T(t)Q\underline{\Xi}\Xi^TQy(t) + Mu(y(t))^T L^2y(t) \\ &\quad + Mvy(t - \tau(t))^T L^2y(t - \tau(t)) \\ &\quad + \frac{M}{v}y^T(t)Q(|\Gamma| + |\Delta|)(|\Gamma| + |\Delta|)^TQy(t) \\ &\leq y^T(t)[-2mQ\underline{D} + \frac{M}{u}Q\underline{\Xi}\Xi^TQ + MuL^2 + \frac{M}{v}Q(|\Gamma| + |\Delta|)(|\Gamma| \\ &\quad + |\Delta|)^TQ]y(t) + Mvy(t - \tau(t))^T L^2y(t - \tau(t)) \\ &= y^T(t)[-2mQ\underline{D} + \frac{M}{u}Q\underline{\Xi}\Xi^TQ + MuL^2 + \frac{M}{v}Q(|\Gamma| + |\Delta|)(|\Gamma| \\ &\quad + |\Delta|)^TQ]y(t) + Mvy(t - \tau(t))^T L^2y(t - \tau(t)) \\ &\leq -wV(y(t)) + y^T(t)Q[-2mP\underline{D} + wP + \frac{M}{u}\underline{\Xi}\Xi^T + MuPL^2P \\ &\quad + \frac{M}{v}(|\Gamma| + |\Delta|)(|\Gamma| + |\Delta|)^T]Qy(t) + Mvy(t - \tau(t))^T L^2y(t - \tau(t)) \\ &\leq -wV(y(t)) + y^T(t)Q[-2mP\underline{D} + MwP + \frac{M}{u}\Xi^* + MuPL^2P \\ &\quad + \frac{M}{v}\Omega^*]Qy(t) + Mvy(t - \tau(t))^T L^2y(t - \tau(t)) \\ &\leq -wV(y(t)) + Mvy(t - \tau(t))^T L^2y(t - \tau(t)) \\ &\leq -wV(y(t)) + Mv \max_{1 \leq i \leq n} \{l_i^2\} y(t - \tau(t))^T y(t - \tau(t)) \\ &= -wV(y(t)) + Mv \max_{1 \leq i \leq n} \{l_i^2\} \|y(t - \tau(t))\|. \end{aligned} \tag{13}$$

By the above inequality (13), we have, for $t > \tau$,

$$\begin{aligned} V(y(t)) &\leq V(y(\tau))e^{-w(t-\tau)} + Mv \max_{1 \leq i \leq n} \{l_i^2\} \int_{\tau}^t e^{-w(t-s)} \|y(s - \tau(s))\|^2 ds \\ &\leq \lambda_M(Q)a_\tau^2 e^{-w(t-\tau)} + Mv \max_{1 \leq i \leq n} \{l_i^2\} \int_{\tau}^t e^{-w(t-s)} \|y(s - \tau(s))\|^2 ds \end{aligned} \quad (14)$$

By Eq. (12) and Eq. (14), we have

$$\begin{aligned} \|y(t)\|^2 &\leq \frac{1}{\lambda_m(Q)} [\lambda_M(Q)a_\tau^2 e^{-w(t-\tau)} \\ &\quad + Mv \max_{1 \leq i \leq n} \{l_i^2\} \int_{\tau}^t e^{-w(t-s)} \|y(s - \tau(s))\|^2 ds] \end{aligned} \quad (15)$$

In order to prove that the origin of system (6) is robustly exponentially stable, it is enough to show that the solution to system (6) has the following property:

$$\|y(t)\| \leq \left(\frac{1}{1 - c^*} \right)^{\frac{1}{2}} K e^{-\frac{\epsilon}{2}(t-\tau)}, \quad t \geq \tau, \quad (16)$$

where $K = \left(\frac{1}{\lambda_m(Q)} [\lambda_M(Q)a_\tau^2 + \frac{M}{w} v \max_{1 \leq i \leq n} \{l_i^2\} a_{2\tau}^2 e^{w\tau}] \right)^{\frac{1}{2}}$ and ϵ is a selected constant satisfying $0 < \epsilon < w$ and $c^* = \frac{1}{\lambda_m(Q)} \frac{1}{w-\epsilon} v \max_{1 \leq i \leq n} \{l_i^2\} e^{\epsilon\tau} < 1$. Notice that the condition (ii) of this theorem implies the existence of ϵ .

Obviously, the inequality (16) is equivalent to the following inequality holding for any $\rho > 1$,

$$\|y(t)\| \leq \rho \left(\frac{1}{1 - c^*} \right)^{\frac{1}{2}} K e^{-\frac{\epsilon}{2}(t-\tau)}, \quad t \geq \tau. \quad (17)$$

From the inequality (15), it is clear that when $t = \tau$, inequality (16) holds. Now, we assume that there exist $t_0 > \tau$ and $\rho_0 > 1$ such that

$$\|y(t_0)\| = \rho_0 \left(\frac{1}{1 - c^*} \right)^{\frac{1}{2}} K e^{-\frac{\epsilon}{2}(t_0-\tau)}, \quad (18)$$

and, for any $t \in [\tau, t_0)$,

$$\|y(t)\| \leq \rho_0 \left(\frac{1}{1 - c^*} \right)^{\frac{1}{2}} K e^{-\frac{\epsilon}{2}(t-\tau)}, \quad (19)$$

Case 1: $t_0 \in (\tau, 2\tau]$. By the inequality (15), we have

$$\begin{aligned} \|y(t_0)\|^2 &\leq \frac{1}{\lambda_m(Q)} [\lambda_M(Q)a_\tau^2 e^{-w(t_0-\tau)} \\ &\quad + Mv \max_{1 \leq i \leq n} \{l_i^2\} \int_{\tau}^{t_0} e^{-w(t_0-s)} \|y(s - \tau(s))\|^2 ds] \end{aligned}$$

$$\begin{aligned}
 &\leq \frac{1}{\lambda_m(Q)} [\lambda_M(Q) a_\tau^2 e^{-\epsilon(t_0-\tau)} + Mv \max_{1 \leq i \leq n} \{l_i^2\} a_{2\tau}^2 \int_\tau^{2\tau} e^{-w(t_0-s)} ds] \\
 &\leq \frac{1}{\lambda_m(Q)} [\lambda_M(Q) a_\tau^2 e^{-\epsilon(t_0-\tau)} \\
 &\quad + \frac{M}{w} v \max_{1 \leq i \leq n} \{l_i^2\} a_{2\tau}^2 (e^{-w(t_0-2\tau)} - e^{-w(t_0-\tau)})] \\
 &= \frac{1}{\lambda_m(Q)} [\lambda_M(Q) a_\tau^2 e^{-\epsilon(t_0-\tau)} + \frac{M}{w} v \max_{1 \leq i \leq n} \{l_i^2\} a_{2\tau}^2 (e^{w\tau} - 1) e^{-w(t_0-\tau)}] \\
 &\leq \frac{1}{\lambda_m(Q)} [\lambda_M(Q) a_\tau^2 e^{-\epsilon(t_0-\tau)} + \frac{M}{w} v \max_{1 \leq i \leq n} \{l_i^2\} a_{2\tau}^2 e^{w\tau} e^{-\epsilon(t_0-\tau)}] \\
 &= K^2 e^{-\epsilon(t_0-\tau)} \\
 &< \frac{\rho_0^2}{1-c^*} K^2 e^{-\epsilon(t_0-\tau)} \tag{20}
 \end{aligned}$$

It is contradicted to the equation (18).

Case 2: $t_0 \in (2\tau, \infty)$. From the inequality (15), we have

$$\begin{aligned}
 \|y(t_0)\|^2 &\leq \frac{1}{\lambda_m(Q)} [\lambda_M(Q) a_\tau^2 e^{-w(t_0-\tau)} + Mv \max_{1 \leq i \leq n} \{l_i^2\} a_{2\tau}^2 \int_\tau^{2\tau} e^{-w(t_0-s)} ds \\
 &\quad + Mv \max_{1 \leq i \leq n} \{l_i^2\} \int_{2\tau}^{t_0} e^{-w(t_0-s)} \|y(s-\tau)\|^2 ds] \\
 &\leq \frac{1}{\lambda_m(Q)} [\lambda_M(Q) a_\tau^2 e^{-\epsilon(t_0-\tau)} + Mv \max_{1 \leq i \leq n} \{l_i^2\} a_{2\tau}^2 \int_\tau^{2\tau} e^{-w(t_0-s)} ds \\
 &\quad + Mv \rho_0^2 \left(\frac{1}{1-c^*} \right) K^2 \max_{1 \leq i \leq n} \{l_i^2\} \times \int_{2\tau}^{t_0} e^{-w(t_0-s)} e^{-\epsilon(s-\tau(s)-\tau)} ds] \\
 &\leq \frac{1}{\lambda_m(Q)} [\lambda_M(Q) a_\tau^2 e^{-\epsilon(t_0-\tau)} + Mv \max_{1 \leq i \leq n} \{l_i^2\} a_{2\tau}^2 \int_\tau^{2\tau} e^{-w(t_0-s)} ds \\
 &\quad + Mv \rho_0^2 \left(\frac{1}{1-c^*} \right) K^2 \max_{1 \leq i \leq n} \{l_i^2\} \int_{2\tau}^{t_0} e^{-w(t_0-s)} e^{-\epsilon(s-2\tau)} ds] \\
 &\leq \frac{1}{\lambda_m(Q)} [\lambda_M(Q) a_\tau^2 e^{-\epsilon(t_0-\tau)} + \frac{1}{w} e^{w\tau} Mv \max_{1 \leq i \leq n} \{l_i^2\} a_{2\tau}^2 e^{-\epsilon(t_0-\tau)} \\
 &\quad + Mv \rho_0^2 \left(\frac{1}{1-c^*} \right) K^2 e^{\epsilon\tau} \max_{1 \leq i \leq n} \{l_i^2\} \frac{1}{(w-\epsilon)} e^{-\epsilon(t_0-\tau)}] \\
 &\leq [K^2 + \frac{\rho_0^2}{1-c^*} K^2 c^*] e^{-\epsilon(t_0-\tau)} \\
 &< \frac{\rho_0^2}{1-c^*} K^2 e^{-\epsilon(t_0-\tau)} \tag{21}
 \end{aligned}$$

It is contradicted to the assumption (18). So far, we have proved that (17) is correct, thus, (16) is correct, i.e., the system (6) is robustly stable. The proof of the theorem is completed

Let $u = v = 1$, then we have the following corollary.

Corollary 1. Suppose that there exist a diagonal matrix $P = \text{diag}(p_1, \dots, p_n) > 0$ and a positive w such that

(i) The LMI holds:

$$0 < \begin{bmatrix} 2mP\underline{D} - wP - M\underline{\Xi}^* - M\underline{\Omega}^* & \sqrt{M}PL \\ \sqrt{M}LP & I_n \end{bmatrix}$$

(ii) $c \equiv \frac{M}{w} \max_{1 \leq i \leq n} \{p_i\} \max_{1 \leq i \leq n} \{l_i^2\} < 1$

where I_n is an n th-order identity matrix, $L = \text{diag}(l_i)_{n \times n}$ and l_i being the Lipschitz constants in (3), $\underline{D}, \underline{\Xi}^*, \underline{\Omega}^*$ defined in (4). Then, model (1) is exponentially robustly stable under the assumption H .

4 Conclusion

In this paper, based on the Lyapunov method and linear matrix inequalities technique, a new criterion for the robust stability of interval fuzzy Cohen-Grossberg neural networks with time-varying delays has been obtained. It is believed that that the robust stability is very important in designing Cohen-Grossberg neural networks. Thus, the results present in this paper are useful in the application and design of neural networks since the conditions are easy to check in practice.

Acknowledgments. This work was supported by the Natural Science Foundation of China under Grants: 60774051 and 10971240, Program for New Century Excellent Talents in Universities of China under Grant NCET-06-0658, the Fok Ying Tung Education Foundation under Grant 111068.

References

1. Cao, J., Liang, J.: Boundedness and stability for Cohen-Grossberg neural network with time-varying delays. *J. of Math. Anal. and Appl.* 296, 665–685 (2004)
2. Chen, T., Rong, L.: Delay-independent stability analysis of Cohen-Grossberg neural networks. *Physics Letters A* 317, 436–449 (2003)
3. Chen, T., Rong, L.: Robust global exponential stability of Cohen-Grossberg neural networks with time delay. *IEEE Transactions on Neural Networks* 15, 203–206 (2004)
4. Cohen, M.A., Grossberg, S.: Absolute stability and global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man and Cybernetics* SMC-13, 815–821 (1983)
5. Jiang, M., Shen, Y., Liao, X.X.: Boundedness and global exponential stability for generalized Cohen-Grossberg neural networks with variable delay. *Applied Mathematics and Computation* 172, 379–393 (2006)
6. Grossberg, S.: Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks* 1, 17–61 (1988)
7. Hopfield, J.J.: Neural networks with graded response have collective computational properties like those of two-stage neurons. *Proc. Nat. Acad. Sci. USA* 81, 3088–3092 (1984)

8. Horn, R.A., Johnson, C.R.: Matrix Analysis, vol. 17. Cambridge University Press, London (1990)
9. Huang, T., Li, C., Chen, G.: Stability of Cohen-Grossberg Neural Networks with Unbounded Distributed Delays. *Chaos, Solitons & Fractals* 34, 992–996 (2007)
10. Huang, T.: Exponential stability of delayed fuzzy cellular neural networks with diffusion. *Chaos, Solitons & Fractals* 31, 658–664 (2007)
11. Huang, T., Chan, A., Huang, Y., Cao, J.: Stability of Cohen-Grossberg neural networks with time-varying delays. *Neural Networks* 20, 868–873 (2007)
12. Li, C., Liao, X.F., Zhang, R.: Global robust asymptotical stability of multi-delayed interval neural networks: an LMI approach. *Physics Letters A* 328, 452–462 (2004)
13. Liao, X.F., Chen, G., Sanchez, E.N.: LMI-based approach for asymptotically stability analysis of delayed neural networks. *IEEE Transactions on CAS-I* 49, 1033–1039 (2002)
14. Liao, X.F., Li, C., Wong, K.W.: Criteria for exponential stability of Cohen-Grossberg neural networks. *Neural Networks* 17, 1401–1414 (2004)
15. Liao, X.F., Yu, J.B.: Robust stability for interval Hopfield neural networks with time delay. *IEEE Trans. Neural Networks* 9, 1042–1045 (1998)
16. Liao, X.X.: Mathematical theory of cellular neural networks (II). *Science in China (A)* 38, 542–551 (1995)
17. Liu, Y., Tang, W.: Exponential Stability of Fuzzy Cellular Neural Networks with Constant and Time-varying Delays. *Physics Letters A* 323, 224–233 (2004)
18. Song, Q., Cao, J.: Dynamical behaviors of discrete-time fuzzy cellular neural networks with variable delays and impulses. *J. Franklin Inst.* 345, 39–59 (2008)
19. Song, Q., Cao, J.: Impulsive effects on stability of fuzzy Cohen-Grossberg neural networks with time-varying delays. *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics* 37, 733–741 (2007)
20. Wang, L., Zou, X.: Exponential stability of Cohen-Grossberg neural networks. *Neural Networks* 15, 415–422 (2002)
21. Yang, T., Yang, L.B., Wu, C.W., Chua, L.O.: Fuzzy cellular neural networks: theory. In: *Proceedings of IEEE International Workshop on Cellular Neural networks and Applications*, pp. 181–186 (1996)
22. Yang, T., Yang, L.B., Wu, C.W., Chua, L.O.: Fuzzy cellular neural networks: applications. In: *Proceedings of IEEE International Workshop on Cellular Neural networks and Applications*, pp. 225–230 (1996)
23. Yang, T., Yang, L.B.: The global stability of fuzzy cellular neural network. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 43, 880–883 (1996)
24. Yuan, K., Cao, J., Li, H.-X.: Robust Stability of Switched Cohen-Grossberg Neural Networks with Mixed Time-Varying Delays. *IEEE Transactions on Systems, Man and Cybernetics, Part B* 36, 1356–1363 (2006)
25. Zeng, Z.G., Wang, J., Liao, X.X.: Stability analysis of delayed cellular neural networks described using cloning templates. *IEEE Trans. Circ. Syst. I* 51, 2313–2324 (2004)
26. Zeng, Z.G., Wang, J.: Global exponential stability of recurrent neural networks with time-varying delays in the presence of strong external stimuli. *Neural Networks* 19, 1528–1537 (2006)
27. Boyd, S., Ghaoui, L.E., Feron, E., Balakrishnan, V.: *Linear Matrix Inequalities in System and Control Theory*. SIAM, Philadelphia (1994)

An Adaptive Threshold in Joint Approximate Diagonalization by the Information Criterion

Yoshitatsu Matsuda¹ and Kazunori Yamaguchi²

¹ Department of Integrated Information Technology,
Aoyama Gakuin University,

5-10-1 Fuchinobe, Sagamihara-shi, Kanagawa, 229-8558, Japan
matsuda@it.aoyama.ac.jp

<http://www-haradalb.it.aoyama.ac.jp/~matsuda>

² Department of General Systems Studies,
Graduate School of Arts and Sciences, The University of Tokyo,
3-8-1, Komaba, Meguro-ku, Tokyo, 153-8902, Japan
yamaguch@graco.c.u-tokyo.ac.jp

Abstract. Joint approximate diagonalization is one of well-known methods for solving independent component analysis and blind source separation. It calculates an orthonormal separating matrix which diagonalizes many cumulant matrices of given observed signals as accurately as possible. It has been known that such diagonalization can be carried out efficiently by the Jacobi method, where the optimization for each pair of signals is repeated until the convergence of the whole separating matrix. The Jacobi method decides whether the optimization is actually applied to a given pair by a convergence decision condition. Generally, a fixed threshold is used as the condition. Though a sufficiently small threshold is desirable for the accuracy of results, the speed of convergence is quite slow if the threshold is too small. In this paper, we propose a new decision condition with an adaptive threshold for joint approximate diagonalization. The condition is theoretically derived by a model selection approach to a simple generative model of cumulants in the similar way as in Akaike information criterion. In consequence, the adaptive threshold is given as the current average of all the cumulants. Only if the expected reduction of the cumulants on each pair is larger than the adaptive threshold, the pair is actually optimized. Numerical results verify that the method can choose a suitable threshold for artificial data and image separation.

Keywords: signal processing, independent component analysis, joint approximate diagonalization, Akaike information criterion.

1 Introduction

Independent component analysis (ICA) is a widely-used method in signal processing [1,2]. It solves blind source separation problems under the assumption that source signals are statistically independent of each other. In the linear model (given as $\mathbf{X} = \mathbf{W}\mathbf{S}$), it estimates the $N \times N$ mixing matrix \mathbf{W} and the source

signals \mathbf{S} from only the observed signals \mathbf{X} . N is the number of signals. Joint approximate diagonalization (called JADE in [3,4]) is one of efficient methods for estimating \mathbf{W} . Now, Δ_{pq} is defined as an $N \times N$ matrix whose (i, j) element is κ_{ijppq} . Here, κ_{ijppq} is the 4-th order cumulants of \mathbf{X} . It is easily proved that $\tilde{\Delta}_{pq} = \mathbf{V} \Delta_{pq} \mathbf{V}'$ is a diagonal matrix for any p and q if \mathbf{V} is the accurate separating matrix. Therefore, \mathbf{W} can be estimated as \mathbf{V} which diagonalizes Δ_{pq} as accurately as possible for many p 's and q 's. Besides, because \mathbf{X} is assumed to be pre-whitened, \mathbf{V} is constrained to an orthonormal matrix. Then, the estimated separating matrix $\hat{\mathbf{V}}$ is given as

$$\hat{\mathbf{V}} = \operatorname{argmin}_{\mathbf{V}} \sum_{p,q \geq p} \sum_{i,j > i,k} (\tilde{\kappa}_{ijppq})^2 \tag{1}$$

where $\tilde{\Delta}_{pq} = (\tilde{\kappa}_{ijppq})$. Though the original JADE algorithm in [3] uses the summation over $\sum_{p,q=p}$ instead of $\sum_{p,q \geq p}$ for reducing computational costs, Eq. (1) is employed in this paper for achieving more accurate results. Because it is relatively difficult to calculate $\hat{\mathbf{V}}$ directly, the Jacobi method is often used. The method optimizes the objective function $\Psi = \sum_{p,q \geq p} \sum_{i,j > i} (\tilde{\kappa}_{ijppq})^2$ only for each pair (i, j) . By sweeping the optimizations over all the pairs repeatedly, the whole \mathbf{V} can be estimated. Because \mathbf{V} is an orthonormal matrix, each pair optimization is given as a 2×2 rotation matrix $(\cos \phi, \sin \phi; -\sin \phi, \cos \phi)$ which has only a single parameter ϕ . Because the optimal $\hat{\phi}$ can be calculated analytically and efficiently, JADE is known to be efficient.

Now, we focus on the decision condition in the Jacobi method. Generally, each pair optimization has to decide whether the ‘‘actual’’ rotation is needed. Only if every pair does not need any actual rotations, the convergence of the whole estimated matrix is declared. The classical Jacobi method employs a simple decision policy using a fixed small threshold ϵ . That is, only if $\hat{\phi} > \epsilon$, the actual rotation is applied. In order to obtain accurate results, an extremely small ϵ has been used in many cases. However, the convergence is quite slow if ϵ is too small. In this paper, an ‘‘optimal’’ decision condition is proposed, which is given by minimizing an information criterion on an approximate probabilistic model of cumulants. Its concrete form is derived in the similar way as in the model selection theory and Akaike information criterion (AIC) [5,6]. Though the information criteria such as AIC are widely used in order to estimate the number of sources [7], they give a criterion for estimating the ‘‘goodness’’ of the whole separating matrix. On the other hand, the proposed method focuses on each pair optimization. In our previous works [8], a fixed threshold was automatically determined by a model selection approach, but the threshold was too large at the convergence phase. In this paper, we propose a new method, which is effective even at the convergence phase. It determines the best threshold adaptively for each pair optimization with little additional costs.

This paper is organized as follows. In Section 2, the model selection theory is introduced and the derivation of AIC is explained in brief. In Section 3, a new

adaptive threshold for the decision is proposed by minimizing an information criterion. Section 4 shows the results of numerical experiments. Lastly, this paper is concluded in Section 5.

2 Model Selection and Akaike Information Criterion

The model selection problem is defined as follows. \mathbf{x} represents samples which are generated from a “true” probabilistic distribution $f(\mathbf{x})$. The purpose is to estimate $f(\mathbf{x})$ as accurately as possible from given samples \mathbf{x}_i through a probabilistic model $g(\mathbf{x}|\boldsymbol{\theta})$. Here, $\boldsymbol{\theta} = (\theta_k)$ is the vector of parameters of the model. One of well-known estimators of $\boldsymbol{\theta}$ is the maximum likelihood estimation (MLE) $\hat{\boldsymbol{\theta}}$, which is given as

$$\hat{\boldsymbol{\theta}} = \operatorname{argmax}_{\boldsymbol{\theta}} L(\mathbf{x}, \boldsymbol{\theta}) \quad (2)$$

where $L(\mathbf{x}, \boldsymbol{\theta}) = \log(g(\mathbf{x}|\boldsymbol{\theta}))$. Given a sufficiently large number of samples, MLE maximize the log-likelihood $E_{\mathbf{x}}(L(\mathbf{x}, \boldsymbol{\theta}))$ ($E_{\mathbf{x}}(\cdot)$ is the expectation operator over $f(\mathbf{x})$). Though MLE is effective for discovering approximately optimal parameters in a single model, it is not useful for comparing multiple models because $E_{\mathbf{x}}(L(\mathbf{x}, \boldsymbol{\theta}))$ can be arbitrarily large by giving a model with many parameters. Akaike information criterion [5] is a method for solving this model selection problem. It employs the following criterion T ,

$$T = E_{\mathbf{y}} \left(E_{\mathbf{x}} \left(L(\mathbf{x}, \hat{\boldsymbol{\theta}}(\mathbf{y})) \right) \right), \quad (3)$$

instead of the log-likelihood. Though \mathbf{x} and \mathbf{y} are given according to the same distribution $f(\mathbf{x})$ (and $f(\mathbf{y})$), they are generated independently. T is consistent with the cross validation method. It is difficult to estimate T accurately. However, if $\hat{\boldsymbol{\theta}}(\mathbf{y})$ is sufficiently close to the vector of true (and optimal) parameters $\boldsymbol{\theta}_o$ (“true” means $g(\mathbf{x}|\boldsymbol{\theta}_o) = f(\mathbf{x})$), the approximation of T can be derived analytically. In consequence, T is approximated as

$$T \simeq E_{\mathbf{x}} \left(L(\mathbf{x}, \hat{\boldsymbol{\theta}}(\mathbf{x})) \right) - K \quad (4)$$

where K is the number of parameters in $\boldsymbol{\theta}$ (see [6] for the details of the derivation). The estimator of $E_{\mathbf{x}} \left(L(\mathbf{x}, \hat{\boldsymbol{\theta}}(\mathbf{x})) \right)$ for a given sample \mathbf{x} is given as $L(\mathbf{x}, \hat{\boldsymbol{\theta}}(\mathbf{x}))$. So, the estimator of T is given as

$$T \simeq L(\mathbf{x}, \hat{\boldsymbol{\theta}}(\mathbf{x})) - K \quad (5)$$

which is the general form of AIC. It tends to prefer a simple model with fewer parameters, so it can suppress the increase of unnecessary parameters of the model. AIC gives a mathematical framework explaining “Occam’s razor.”

3 Adaptive Threshold for Pair Optimization

Each term $(\tilde{\kappa}_{ijkk})^2$ in the JADE objective function Ψ is denoted by $c_p \in \mathbf{c}$ ($p = 1, \dots, M = N^2(N-1)/2$). Assume that the current \mathbf{V} is near to the optimum. The key idea in our proposed approach is to regard each c_p as an independent sample from a generative probabilistic model $g(c_p)$. In general, a square cumulant c_p depends on c_q each other. However, at the optimum, each c_p is approximated as an independent noise to the ideal value “0”. Therefore, it is assumed here that every c_p are independent of each other. Because c_p is non-negative, the following exponential distribution is employed as the generative model:

$$g(c_p|\lambda) = \frac{e^{-\frac{c_p}{\lambda}}}{\lambda} \tag{6}$$

where λ is a parameter and is equal to the mean of the distribution. Then, the log-likelihood w.r.t. the rotation ϕ for a given pair (i, j) is given as

$$\begin{aligned} \sum_{c_p \in \mathbf{c}} L(c_p, \phi, \lambda) &= \sum_{i,j>i,k} \left(-\log(\lambda) - \frac{(\tilde{\kappa}_{ijkk}(\phi))^2}{\lambda} \right) \\ &= -M \log(\lambda) - \frac{\Psi(\phi)}{\lambda}. \end{aligned} \tag{7}$$

Therefore, MLE $\hat{\phi}$ on this log-likelihood is completely equivalent to the optimal $\hat{\phi}$ for each pair optimization of Ψ in JADE (see Section 4). It shows that the proposed simple generative model is consistent with the JADE algorithm. Now, AIC of this generative model is utilized for deciding whether the actual rotation of $\hat{\phi}$ is needed for each pair in JADE. For this purpose, AIC of the current state is compared with that of the state after the actual rotation. Only if AIC increases by the rotation, the actual rotation is preferable. The current information criterion T_{curr} before the actual rotation is given as

$$T_{\text{curr}} = E_{\mathbf{c}} \left(E_{\mathbf{c}'} \left(L \left(c_p \in \mathbf{c}, \phi = 0, \hat{\lambda}_{\text{curr}}(\mathbf{c}') \right) \right) \right) \tag{8}$$

where $c_p \in \mathbf{c}$ and $c'_p \in \mathbf{c}'$ are generated independently according to the same distribution, and $\hat{\lambda}_{\text{curr}}$ (MLE of λ) is determined by maximizing Eq. (7):

$$\hat{\lambda}_{\text{curr}} = \frac{\Psi(\phi = 0)}{M} \tag{9}$$

which corresponds to the mean of the current square cumulants. Because the free parameter is only λ , T_{curr} is approximated as

$$T_{\text{curr}} \simeq L \left(c_p \in \mathbf{c}, \phi = 0, \hat{\lambda}_{\text{curr}}(\mathbf{c}) \right) - 1, \tag{10}$$

where $K = 1$ in Eq. (5). On the other hand, the information criterion after the rotation (T_{rot}) is given as

$$T_{\text{rot}} = E_{\mathbf{c}} \left(E_{\mathbf{c}'} \left(L \left(c_p \in \mathbf{c}, \hat{\phi}(\mathbf{c}'), \hat{\lambda}_{\text{rot}}(\mathbf{c}') \right) \right) \right) \tag{11}$$

where $\hat{\lambda}_{\text{rot}}$ is given as

$$\hat{\lambda}_{\text{rot}} = \frac{\Psi(\hat{\phi})}{M}, \tag{12}$$

which corresponds to the mean of the square cumulants after the rotation. Because ϕ and λ are the free parameters, T_{rot} is approximated as

$$T_{\text{rot}} \simeq L \left(c_p \in \mathbf{c}, \hat{\phi}(\mathbf{c}), \hat{\lambda}_{\text{rot}}(\mathbf{c}) \right) - 2, \tag{13}$$

where $K = 2$. The actual rotation should be done only if $\delta T = T_{\text{rot}} - T_{\text{curr}} > 0$. By Eq. (7), δT is transformed into

$$\begin{aligned} \delta T &= -M \log(\lambda_{\text{rot}}) - \frac{\Psi(\hat{\phi})}{\lambda_{\text{rot}}} - 2 \\ &+ M \log(\lambda_{\text{curr}}) + \frac{\Psi(0)}{\lambda_{\text{curr}}} + 1. \end{aligned} \tag{14}$$

It is shown by Eqs. (9) and (12) that $\frac{\Psi(\hat{\phi})}{\lambda_{\text{rot}}}$ and $\frac{\Psi(0)}{\lambda_{\text{curr}}}$ are the same constant M . Then, δT is transformed further into

$$\delta T = -M \log\left(\frac{\lambda_{\text{rot}}}{\lambda_{\text{curr}}}\right) - 1. \tag{15}$$

Now, $\delta\Psi(\hat{\phi}) = \Psi(0) - \Psi(\hat{\phi})$ is introduced, which corresponds to the reduction of Ψ by the rotation $\hat{\phi}$. Then, $\log\left(\frac{\lambda_{\text{rot}}}{\lambda_{\text{curr}}}\right)$ is rewritten as

$$\log\left(\frac{\lambda_{\text{rot}}}{\lambda_{\text{curr}}}\right) = \log\left(\frac{\Psi(\hat{\phi})}{\Psi(0)}\right) = \log\left(1 - \frac{\delta\Psi(\hat{\phi})}{\Psi(0)}\right). \tag{16}$$

Because $\delta\Psi$ is related to only the terms on i and j , it is generally quite smaller than the total summation Ψ if N is large. So, by assuming that $\delta\Psi(\hat{\phi}) \ll 1$, the following approximation holds:

$$\log\left(1 - \frac{\delta\Psi(\hat{\phi})}{\Psi(0)}\right) \simeq -\frac{\delta\Psi(\hat{\phi})}{\Psi(0)}. \tag{17}$$

Thus, $\delta T > 0$ is given as

$$M \frac{\delta\Psi(\hat{\phi})}{\Psi(0)} - 1 > 0, \tag{18}$$

which is rewritten as the following final decision condition:

$$\delta\Psi(\hat{\phi}) > \frac{\Psi(0)}{M}. \tag{19}$$

It means that the actual rotation is necessary if the reduction of Ψ by $\hat{\phi}$ is larger than the mean of the current square cumulants. It gives an adaptive threshold $\frac{\Psi}{M}$, which is smaller when the current state is closer to the optimum. Regarding computational costs, the costs of each estimation of $\hat{\phi}$ are only $\frac{1}{N}$ as high as those of each actual rotation. Besides, the costs of calculating the total value of Ψ is negligible because Ψ is easily updated at each actual rotation [3]. Therefore, the costs of decisions are much less than those of actual rotations if N is large.

4 Results

Here, JADE with the proposed adaptive threshold is compared with the previous fixed ones ϵ in blind source separation of artificial data and an image separation problem. Regarding artificial data, The number of sources was set to 20 and 40 ($N = 20, 40$). A half of the sources were generated by the Laplace distribution (super-Gaussian) and the other half by the uniform distribution (sub-Gaussian). The number of samples was set to 50000, and the mixing matrix \mathbf{W} was randomly generated. Regarding the image separation, the sources were 12 grayscale images of 256×256 pixels from the USC-SIPI database and a 12×12 mixing matrix was given randomly, where $N = 12$ and the number of samples is 65536.

Fig. 1 shows the decreasing curves of Amari's separating errors [9] along the number of the actual rotations by the adaptive threshold and fixed ones ($\epsilon = 10^{-1}, 10^{-2},$ and 10^{-6}). The separating error is defined as the sum of normalized non-diagonal elements of the product of the estimated separating matrix and the given mixing one. If the error is equal to 0, the estimated separating matrix is equivalent to the inverse of the mixing one except for scaling factors. They were averaged over 10 runs. It shows that the method with the adaptive threshold converged to the results which are almost equivalent to those with a sufficiently small threshold $\epsilon = 10^{-6}$. In addition, the convergence speed was relatively faster than or almost equivalent to the results with $\epsilon = 10^{-2}$. Therefore, the result shows that the adaptive threshold is suitably chosen. In order to inspect more closely the results at the convergence, Fig. 2 shows the trade-off curves of the final error and the number of actual rotations until convergence. They were calculated by numerical experiments with various fixed thresholds from $10^{-\frac{1}{4}}$ to 10^{-6} . It shows that the final error decreases and the number of actual rotations increases as the fixed threshold decreases. It also shows that the error is approximately a constant once the fixed threshold is below a critical point. The results of the adaptive threshold are shown by the black dots in the figures. Surprisingly, they are placed nearly at the critical points where the decreasing curves of the final error converge to the optimal values. In other words, the adaptive threshold gives approximately the most accurate estimation of the separating matrix by the minimum number of actual rotations. Fig. 3 shows the trade-off curve on the image separation problem. The adaptive threshold gives a result near to the critical point for this practical application also. Those results verify the effectiveness of the adaptive threshold.

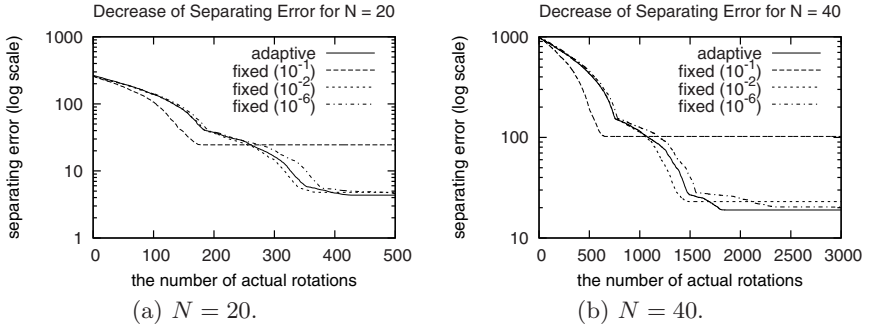


Fig. 1. Decreasing curves of separating errors on a log scale along the number of the actual rotations ($N = 20$ and 40) for artificial data. Solid curves: adaptive threshold. Dashed: fixed threshold $\epsilon = 10^{-1}$. Dotted: $\epsilon = 10^{-2}$. Dot-dashed: $\epsilon = 10^{-6}$.

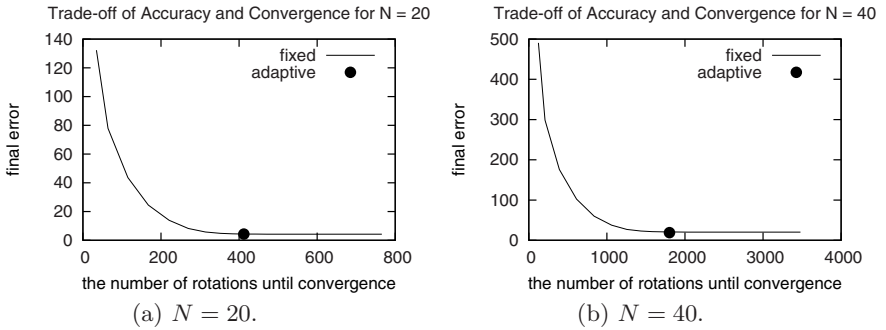


Fig. 2. Comparison of the final error with the number of rotations until convergence for artificial data: the curves show the trade-off between the accuracy and the required number of actual rotations. The curves were calculated by changing fixed threshold ϵ gradually from 10^{-4} to 10^{-6} . The black dots show the results of the adaptive threshold.

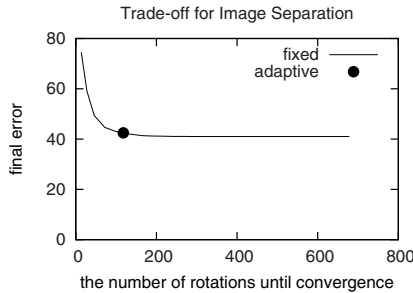


Fig. 3. Comparison of the final error with the number of rotations until convergence for image separation: the settings are the same as in Fig. 2.

5 Conclusion

In this paper, we propose a new decision condition for each pair optimization in the Jacobi method of joint approximate diagonalization. The condition is based on an adaptive threshold, which is derived theoretically by applying a model selection approach to a simple generative model. In consequence, the threshold is equivalent to the mean of the current square cumulants, and it is calculated easily. Numerical results verified that the proposed method could choose the optimal threshold for artificial data and an image separation problem.

In this paper, some experiments on artificial data and image separation were carried out. We are going to apply the proposed method to other practical data. Besides, the proposed approach is expected to be applicable to many cumulants-based algorithms as well as JADE. So, we are going to apply this approach to other algorithms in blind source separation. In addition, it has been known that the model selection theory is effective for solving the overlearning problems. So, we are planning to compare the proposed method with other method in such overlearning problems [10]. This work is partially supported by Grant-in-Aid for Young Scientists (KAKENHI) 19700267.

References

1. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Wiley, Chichester (2001)
2. Cichocki, A., Amari, S.: Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications. Wiley, Chichester (2002)
3. Cardoso, J.-F., Souloumiac, A.: Blind beamforming for non Gaussian signals. IEE Proceedings-F 140(6), 362–370 (1993)
4. Cardoso, J.-F.: High-order contrasts for independent component analysis. Neural Computation 11(1), 157–192 (1999)
5. Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control 19(6), 716–723 (1974)
6. Burnham, K.P., Anderson, D.R.: Model selection and multimodel inference: A practical-theoretic approach, 2nd edn. Springer, Berlin (2002)
7. Wax, M., Kailath, T.: Detection of signals by information theoretic criteria. IEEE Transactions on Acoustics Speech and Signal Processing 33, 387–392 (1985)
8. Matsuda, Y., Yamaguchi, K.: Joint approximate diagonalization utilizing aic-based decision in the jacobi method. In: Alippi, C., et al. (eds.) ICANN 2009, Part II. LNCS, vol. 5769, pp. 135–144. Springer, Heidelberg (2009)
9. Amari, S., Cichocki, A.: A new learning algorithm for blind signal separation. In: Touretzky, D., Mozer, M., Hasselmo, M. (eds.) Advances in Neural Information Processing Systems, vol. 8, pp. 757–763. MIT Press, Cambridge (1996)
10. Särelä, J., Vigário, R.: Overlearning in marginal distribution-based ica: Analysis and solutions. Journal of Machine Learning Research 4, 1447–1469 (2003)

PPoSOM: A Multidimensional Data Visualization Using Probabilistic Assignment Based on Polar SOM

Yang Xu, Lu Xu, Tommy W.S. Chow, and Anthony S.S. Fong

Department of Electronic Engineering, City University of Hong Kong
83 Tat Chee Avenue, Kowloon, Hong Kong

Abstract. A new algorithm named probabilistic polar self-organizing map (PPoSOM) is proposed. PPoSOM is a new variant of Polar SOM which is constructed on 2-D polar coordinates. Data weight and feature are represented by two variables that are radius and angle. The neurons on the map are set as data characteristic benchmarks. Projected data points are trained to get close to the neurons with the highest similarities, while weights of neurons are updated by a probabilistic data assignment method. Thus, not only similar data are gathered together, data characteristics are also reflected by their positions on the map. Our obtained results are compared with conventional SOM and ViSOM. The comparative results show that PPoSOM is a new effective method for multidimensional data visualization.

Keywords: SOM, ViSOM, probabilistic polar SOM (PPoSOM), visualization.

1 Introduction

Data visualization, which is the graphical presentation of data information, has been widely used to solve many problems, e.g. signal compression, pattern recognition, image processing, etc. Principal component analysis (PCA) [1] and multidimensional scaling (MDS) [2] are two classical methods for data reduction and visualization. PCA is an effective method of linear reduction, but it is not suitable for highly nonlinear data. MDS is capable of preserving data structure and inter-point distances, but its computational complexity is heavy and it requires re-computation when new data points are added.

Self-organizing map (SOM) [3-5], a widely used visualization method proposed by Kohonen, is an unsupervised learning neural network to visualize high-dimensional data in a low-dimensional map. SOM is able to present the data topology by assigning each datum to a neuron with the highest similarity. But because of the uniform map grid, SOM cannot preserve the data relationship between clusters or within one cluster. Also, the requirement of pre-defining the map size is another disadvantage of SOM. Visualization-induced SOM (ViSOM)

[6-8] and Probabilistic Regularized SOM (PRSOM) [9] hybridize SOM and MDS in order to preserve the data topology as well as the inter-neuron distances. But they have the same disadvantage as SOM.

In this paper, a new algorithm, Probabilistic Polar SOM (PPoSOM), is proposed for data visualization. It is derived from the concept of a new polar structure [10] with a probabilistic assignment. Instead of Cartesian coordinates, PPoSOM visualizes data in a 2-D polar coordinates map with two variables: radius and angle. These two variables represent data weight and feature respectively. The neurons learn data feature by a probabilistic data assignment method in [9], and the projected data points approach the neurons with similar features. As a result, the data topology as well as the inter-data distance is preserved. Simulation results and the comparisons with SOM and ViSOM show that PPoSOM exhibits remarkable performance on data visualization.

2 Background

2.1 Self-Organizing Map (SOM)

SOM [3-5] consists of N neurons on a low-dimensional map, usually a 2-D grid. Each neuron j , which has the same d -dimensions as input data, is denoted by $w_j = (w_{j1}, w_{j2}, \dots, w_{jd})^T$. During each training process, an input datum x_i is randomly chosen from the input space. The winning neuron c which weight vector is the most similar to this datum is determined by

$$c = \arg \min_j \|x_i - w_j\|, j \in \{1, \dots, N\}. \quad (1)$$

The neighborhood function of winning neuron c , taken as a Gaussian function, is defined by

$$h_{jc}(t) = \exp\left(-\frac{\|Pos_j - Pos_c\|^2}{2\sigma(t)^2}\right), \quad j \in N_c. \quad (2)$$

where N_c is the neighboring set of the winning neuron c , Pos_j and Pos_c are the coordinates of neuron j and c respectively.

The weight updating formula is

$$w_j(t+1) = \begin{cases} w_j(t) + \varepsilon(t)h_{jc}(t)(x_i(t) - w_j(t)), & \forall j \in N_c \\ w_j(t), & otherwise \end{cases} \quad (3)$$

Both the learning rate $\varepsilon(t)$ and the neighborhood $\sigma(t)$ monotonically decrease with time.

After training, similar input data are projected onto adjacent neurons on the output map. But several input data may be projected onto a single neuron making data relationship between clusters or within one cluster difficult to be preserved.

2.2 Visualization-Induced SOM (ViSOM)

ViSOM [6-8] is proposed to preserve the inter-data distances as well as the data topology. ViSOM use the similar map structure as SOM, but the training of winning neuron's neighbors is different. Weight updating formula in ViSOM is defined by

$$w_j(t+1) = w_j(t) + \varepsilon(t)h_{jc}(t)((x(t) - w_c(t) + (w_c(t) - w_j(t))\frac{d_{cj} - \Delta_{cj}\lambda}{\Delta_{cj}\lambda})), j \in N_c \quad (4)$$

where d_{cj} and Δ_{cj} are the distances between nodes c and j in the input space and output space respectively. λ is a positive pre-specified resolution parameter.

ViSOM decomposes the updating force $F_{jx} = x(t) - w_j(t)$ into two forces: $F_{jx} = [x(t) - w_c(t)] + [w_c(t) - w_j(t)] = F_{cx} + F_{cj}$. F_{cx} is the updating force from the winning neuron c to the input data x ; F_{cj} is a lateral contraction force bringing neighboring neuron j to the winner c . The second force regularizes the inter-neuron distance in the output space to resemble that in the input space. But ViSOM has the same drawback as SOM, that some input data points are mapped on the same neuron, making the relationship of these data difficult or even impossible to be preserved.

3 Probabilistic Polar Self-organizing Map (PPoSOM)

PPoSOM is a new self-organizing algorithm designed to provide better visualization. It is constructed on 2-D polar coordinates, and the projected data points on the map are expressed by two variables: angle and radius, representing the data feature and weight respectively. The whole circular map is divided into different angles and radii. Each angle represents an attribute of the data feature and the radius is related to the data weight. Neurons on the map are set as benchmarks of data characteristics. Their weight initializations are determined by their positions on the map in a way that the angle represents the most significant attribute of the neuron weight and the radius reflects the weight value. The PPoSOM employs a kind of probabilistic assignment [9] which connects an input datum to a neuron with a certain probability. The noised probabilistic assignment $p_i(x(t))$ of neuron i is introduced as follows, and the term "noised" means that $p_i(x(t))$ is affected by probabilistic assignments of neighboring neurons.

$$p_i(x(t)) = \sum_{j=1}^N h_{ij}P_j(x(t)) \quad (5)$$

where $P_j(x(t))$ is the probabilistic assignment of neuron j for input $x(t)$, and h_{ij} is a neighborhood constant satisfying $\sum_{j=1}^N h_{ij} = 1$. They can be taken as:

$$P_j(x(t)) = \frac{1}{C} \left(\frac{1}{\left\| \sum_{k=1}^N h_{jk}(x(t) - w_k) \right\|^2} \right) \tag{6}$$

$$h_{ij} = \frac{\exp(-\frac{\|Pos_i - Pos_j\|^2}{2\sigma^2})}{\sum_{k=1}^N \exp(-\frac{\|Pos_i - Pos_k\|^2}{2\sigma^2})} \tag{7}$$

where C is a normalization constant and the neighborhood radius σ is a constant. A probabilistic data assignment makes PPoSOM effective for dimension reduction and visualization.

Define $N_r(i)$ and $N_\alpha(i)$ as neuron i 's radius and angle respectively, and r_x and α_x as an output datum x 's radius and angle respectively. The executing steps of PPoSOM are as follows:

Step 1. Initialize each neuron according to its position. Normalize the input data, and initialize the polar coordinates of their corresponding output data by setting the radii proportion to their weights and the angles with random values.

Step 2. Randomly select an input datum x and find the winning neuron according to Eq. (1). Update the weights of all neurons by

$$w_i(t+1) = \begin{cases} w_i(t) + \eta_1(x(t) - w_i(t)), & \text{for } i = c \\ w_i(t) + \frac{\eta_2}{M} \sum_{k=1}^M p_i(x(k))(x(k) - w_i(i)), & \text{otherwise} \end{cases} \tag{8}$$

where η_1 and η_2 are the constant learning rates, M is the number of input data.

Step 3. Update the polar coordinates of this datum.

$$r_x(t+1) = r_x(t) + \beta_1(t)(N_r(c) - r_x(t)) \tag{9}$$

$$\alpha_x(t+1) = \alpha_x(t) + \beta_2(t)(N_\alpha(c) - \alpha_x(t)) \tag{10}$$

where β_1 and β_2 are the learning rates that monotonically decrease with time.

Step 4. If the iteration is not over, then go to Step 2. Otherwise, the neural network which exhibits precise data characteristics is obtained.

Upon the completion of training process, the visualization map is created so that each input datum is represented by a radius and an angle on the map. Data with similar features are grouped together, and their common feature is also reflected by their positions on the map. Since the data are not projected onto neurons, the data relationships between different clusters and within one cluster are preserved. Also, PPoSOM does not require re-computation when new instances are presented to it, since there are the benchmark neurons representing data characteristics.

4 Simulation Results

We use two synthetic data sets, iris data set [11] and wine data set [12] to illustrate the advantages of PPOsOM. The visualization results are compared with SOM and ViSOM. The map size of SOM is 20×20 , the number of iterations is 1000 and the learning rate monotonically decreases from 1 to 0.018 with time. The neighborhood range also monotonically decreases from 14.78 to 2. In ViSOM, the map size is the same as that of SOM, and λ is set to 0.1. In PPOsOM, η_1 and η_2 are set to 0.05 and 0.1 respectively.

4.1 Three-Dimensional Synthetic Data Sets

In order to demonstrate the characteristics of representing data with radii and angles, two types of 3-D synthetic data sets are used in this section. Each of them consists of two classes named Class 1 and Class 2, and each class is formed by 100 three-dimensional data points.

In the first data set, the mean vector weights of two classes are $[0.45 \ 0.54 \ 0.54]^T$ and $[2.48 \ 2.48 \ 2.52]^T$ respectively. The data weights in Class 1 are smaller than those in Class 2. The simulation results of PPOsOM, SOM and ViSOM are presented in Fig. 1.

As shown in Fig. 1, Class 1 and Class 2 are well separated from each other in PPOsOM, SOM and ViSOM. In PPOsOM map Fig. 1(a), the radii of the

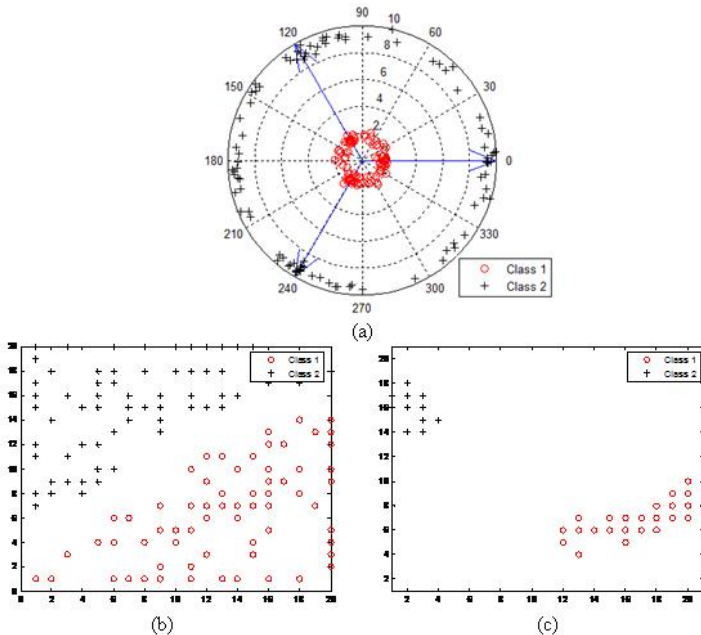


Fig. 1. Visualization of the first 3-D synthetic data set. (a) PPOsOM. (b) SOM. (c) ViSOM.

projected data from Class 1 are smaller than those from Class 2. This is in the agreement with the fact that the average data weight of Class 1 is smaller than that of Class 2. Besides, that evenly distributed data angles indicates the attributes in every data are similar. However, these important characteristics cannot be exhibited in SOM and ViSOM.

In the second data set, the second attributes in Class 1 and the third attributes in Class 2 are larger than the rest. Their mean vectors are $[0.48 \ 2.49 \ 0.54]^T$ and $[0.44 \ 0.50 \ 2.49]^T$ respectively. The visualizations of PPoSOM, SOM and ViSOM are shown in Fig. 2.

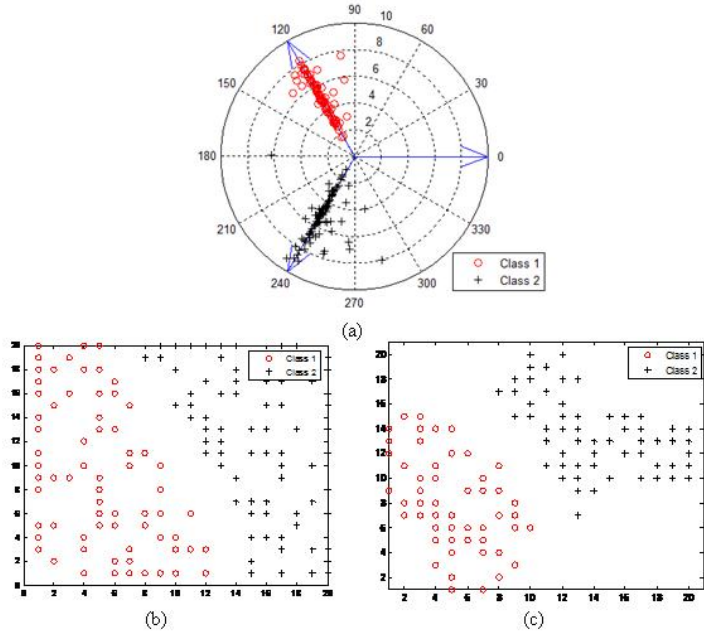


Fig. 2. Visualization of the second 3-D synthetic data set. (a) PPoSOM. (b) SOM. (c) ViSOM.

In Fig. 2, it shows that the two classes are well separated in all the three algorithms. In PPoSOM, data from Class 1 and data from Class 2 are located around 120 degree and 240 degree respectively, representing the significantly large value in the second and the third attribute respectively. It is worth noting that this important characteristic cannot be provided by SOM and ViSOM.

4.2 Iris Data Set

Iris data set [11], one of the well known benchmark data sets for pattern recognition, consists of 3 classes of iris plants, each classes has 50 four-dimensional instances. The first class is clearly separated from the other two. These two are

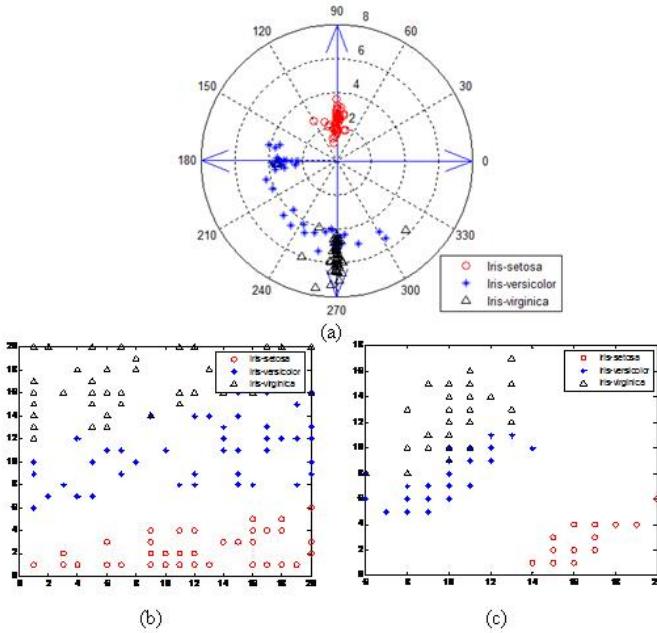


Fig. 3. Visualization of iris data set. (a) PPOsOM. (b) SOM. (c) ViSOM.

overlapped in some extent and not linearly separable from each other. The mean vectors of these three classes are $[0.20 \ 0.59 \ 0.08 \ 0.06]^T$, $[0.45 \ 0.32 \ 0.55 \ 0.51]^T$ and $[0.64 \ 0.41 \ 0.77 \ 0.80]^T$ respectively after normalization.

The visualization results of PPOsOM, SOM and ViSOM are shown in Fig. 3. The characteristics of iris data are clearly shown in Fig. 3(a). The average radius of the first class is the smallest, and that of the third class is the largest. In addition, Fig. 3(a) illustrates different significant attributes in the three classes: the second attribute in Class 1, the third attribute in Class 2 and the fourth attribute in Class 3. The visualization is in agreement with the iris data characteristics. However, the above characteristics cannot be shown in SOM and ViSOM.

4.3 Wine Data Set

The wine data set [12] consists of 178 13-D data points which are divided into three classes. The number of data points in each class is 59, 71 and 48 respectively. These three classes are not well separated.

The visualization results of PPOsOM, SOM and ViSOM are presented in Fig. 4. The characteristics of wine data set are clearly shown in Fig. 4(a). The average radius in Class 1 is the largest, and the average radii in Class 2 and Class 3 are similar. It means that the data weight of Class 1 is larger than other two classes.

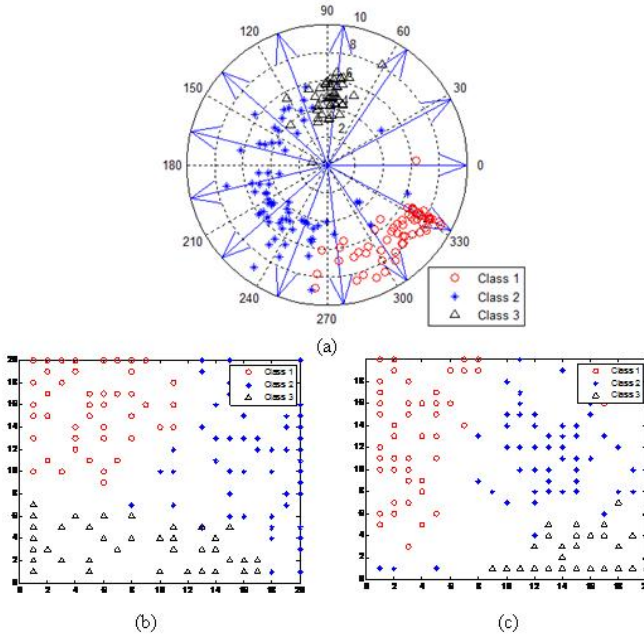


Fig. 4. Visualization of wine data set. (a) PPoSOM. (b) SOM. (c) ViSOM.

Moreover, PPoSOM demonstrates different significant attributes in three classes by angular coordinate. However, in Fig. 4(b) and (c), SOM and ViSOM are incapable to exhibit these data characteristics.

Our results show that the PPoSOM not only can group similar data, it can also make use of the data positions to reflect the characteristics. In other words, PPoSOM is capable of preserving data topology and exhibiting data characteristics. Compared with SOM and ViSOM, which map data on Cartesian coordinates by using Euclidian distance as the only variable, PPoSOM can manifest more precise data characteristics.

5 Conclusion

In this paper, a new self-organizing map called Probabilistic Polar SOM (PPoSOM) is developed for providing a new type of visualization. The design of the PPoSOM was motivated by exhibiting precise data characteristics. Each datum is represented by radius and angle on the polar coordinates. These two variables reflect data weight and feature respectively. Compared with the traditional algorithms which only use Euclidian distance as the variable, PPoSOM provides more characteristics of data. Based on the simulation results, it has been shown that PPoSOM is effective to obtain better visualization, while maintains the topology preservation property.

References

1. Johnson, R.A., Wichern, D.W.: Applied Multivariate Statistical Analysis. Prentice-Hall, Englewood Cliffs (1992)
2. Shepard, R.N., Carroll, J.D.: Parametric representation of nonlinear data structures. In: Krishnaiah, P.R. (ed.) Proc. Int. Symp. Multivariate Anal., pp. 561–592. Academic, New York (1965)
3. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (1997)
4. Kohonen, T.: Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43, 59–69 (1982)
5. Vesanto, J., Alhoniemi, E.: Clustering of the self-organizing map. *IEEE Trans. Neural Networks* 11, 586–600 (2000)
6. Yin, H.: ViSOM: A novel method for multivariate data projection and structure visualization. *IEEE Trans. Neural Networks* 13, 237–243 (2002)
7. Yin, H.: Data visualization and manifold mapping using the ViSOM. *IEEE Trans. Neural Networks* 15, 1005–1016 (2002)
8. Yin, H.: Self-Organizing Maps: Statistical Analysis, Treatment and Applications, PhD Thesis, Department of Electronics, University of York, UK (1996)
9. Wu, S., Chow, T.W.S.: PRSOM: A New Visualization Method by Hybridizing Multidimensional Scaling and Self-Organizing Map. *IEEE Trans. Neural Networks* 16, 1362–1380 (2005)
10. Xu, L., Xu, Y., Chow, T.W.S.: PolSOM: A New Method for Multidimensional Data Visualization, submitted to *Pattern Recognition*
11. Fisher, R.A.: The use of multiple measure in taxonomic problems. *Ann. Eugenics (Part II)* 7, 179–188 (1936)
12. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: Uci Repository of Machine Learning Databases, University of California, Irvine, Dept. of Information and Computer Sciences (1998), <http://archive.ics.uci.edu/ml/>

Slice Oriented Tensor Decomposition of EEG Data for Feature Extraction in Space, Frequency and Time Domains

Qibin Zhao^{1,*}, Cesar F. Caiafa^{1,**}, Andrzej Cichocki¹,
Liqing Zhang², and Anh Huy Phan¹

¹ Laboratory for Advanced Brain Signal Processing, Brain Science Institute,
RIKEN, Saitama, Japan
qbzhao@brain.riken.jp

<http://www.bsp.brain.riken.jp/>

² MOE-Microsoft Laboratory for Intelligent Computing and Intelligent Systems,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University Shanghai, China

Abstract. In this paper we apply a novel tensor decomposition model of SOD (slice oriented decomposition) to extract slice features from the multichannel time-frequency representation of EEG signals measured for MI (motor imagery) tasks in application to BCI (brain computer interface). The advantages of the SOD based feature extraction approach lie in its capability to obtain slice matrix components across the space, time and frequency domains and the discriminative features across different classes without any prior knowledge of the discriminative frequency bands. Furthermore, the combination of horizontal, lateral and frontal slice features makes our method more robust for the outlier problem. The experiment results demonstrate the effectiveness of our method.

Keywords: Tensor decomposition, EEG, BCI.

1 Introduction

Tensors (also known as n-way arrays) are used in a variety of applications ranging from neuroscience and psychometrics to chemometrics [1-3]. From a viewpoint of data analysis, tensor decomposition is very attractive because it takes into account spatial and temporal correlations between variables more accurately than 2D matrix factorizations, and it usually provides sparse common factors or hidden components with physiological meaning and interpretation. In most applications, especially in neuroscience (EEG, fMRI), the standard PARAFAC and Tucker models were used [4-6].

Feature extraction for high dimension data and high noise data plays an important role in machine learning and pattern recognition. In the real world,

* Corresponding author.

** On leave from Engineering Faculty, University of Buenos Aires, Argentina.

the extracted feature of an object often has some specialized structures and such structures are in the form of 2nd or even higher-order tensor. Recently, multilinear algebra, the algebra of high-order tensors, was applied for analyzing the multifactor structure image ensembles, EEG signals [7] and etc. These methods, such as tensor PCA [8], tensor LDA [9, 10], tensor subspace analysis [11–13], treat original data as second- or high-order tensors. For supervised feature classification [14], the tensor factorization can lead to structured dimensionality reduction by learning multiple interrelated subspaces. In the most existing tensor decomposition models, high-dimension tensors are decomposed to many rank-1 vector components on each mode. Unlike most existing models such as PARAFAC, Tucker and HOSVD, our SOD model is to represent a 3D tensor by outer product of slice matrices and corresponding vectors on each tensor mode rather than rank-1 components. Therefore, the structure of tensor data associated to its horizontal, lateral and frontal slices can be captured. Based on the SOD model, we developed a feature extraction framework for single-trial EEG classification.

This paper is organized as follows: in section 2, SOD model and its main properties are introduced briefly, then the feature extraction framework based on SOD are proposed; in section 3, data analysis results on EEG data are presented and discussed; in section 4, the main conclusions and future perspectives of improvement are presented.

2 Method

2.1 SOD Model

In [15], the Slice Oriented Decomposition (SOD) model was recently proposed as a decomposition method of 3-way tensors that captures the structure of data slices providing also a compact representation. SOD takes into account the interactions among the three modes of a tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{I \times J \times K}$ by decomposing it as a sum of elemental (simple) tensors:

$$\hat{\underline{\mathbf{Y}}} = \sum_{p=1}^P \underline{\mathbf{H}}_p + \sum_{q=1}^Q \underline{\mathbf{L}}_q + \sum_{r=1}^R \underline{\mathbf{F}}_r = \sum_{p=1}^P \underline{\mathbf{H}}_p \circ_1 \mathbf{u}_p + \sum_{q=1}^Q \underline{\mathbf{L}}_q \circ_2 \mathbf{v}_q + \sum_{r=1}^R \underline{\mathbf{F}}_r \circ_3 \mathbf{w}_r, \quad (1)$$

where matrices $\underline{\mathbf{H}}_p$, $\underline{\mathbf{L}}_q$ and $\underline{\mathbf{F}}_r$ are called *matrix components*, vectors \mathbf{u}_p , \mathbf{v}_q and \mathbf{w}_r are called *vector components*, $\underline{\mathbf{H}}$, $\underline{\mathbf{L}}$, $\underline{\mathbf{F}} \in \mathbb{R}^{I \times J \times K}$ and \circ_n is the *n -mode outer product* ($n = 1, 2$ or 3) defined as follows:

$$[\underline{\mathbf{H}}]_{ijk} = [\underline{\mathbf{H}} \circ_1 \mathbf{u}]_{ijk} = h_{jk} u_i, \quad (2)$$

$$[\underline{\mathbf{L}}]_{ijk} = [\underline{\mathbf{L}} \circ_2 \mathbf{v}]_{ijk} = l_{ik} v_j, \quad (3)$$

$$[\underline{\mathbf{F}}]_{ijk} = [\underline{\mathbf{F}} \circ_3 \mathbf{w}]_{ijk} = f_{ij} w_k. \quad (4)$$

The effect of the n -mode outer product is to create simple tensors where slices are scaled versions of a basic matrix. In Fig. 1(a) the equation (1) is illustrated while in Fig. 1(b) the SOD compact representation is shown.

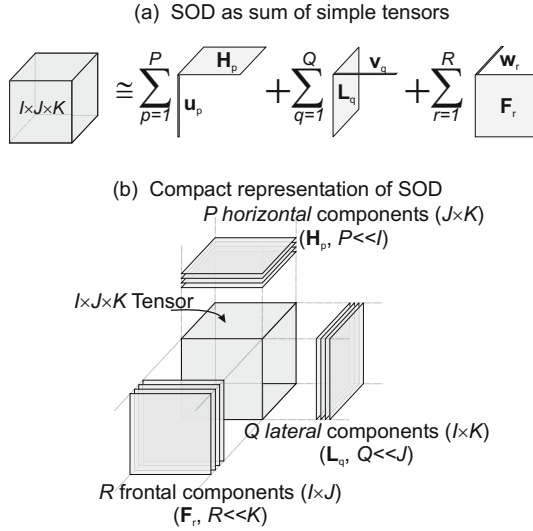


Fig. 1. Slice Oriented Decomposition (SOD) model

When vector and matrix components are constrained to be nonnegative we arrive to the Non-negative SOD (NN-SOD) for which an Alternate Least Squared (ALS) Newton based algorithm is available [15].

2.2 Feature Extraction

To apply SOD for extracting slice features along horizontal, lateral and frontal directions, the class-averaged EEG tensor data $\mathbf{Y}^c, c \in \{1, 2\}$ were decomposed according to Eq.(II) and reorganized as

$$\mathbf{Y}^c = \mathbf{H}^c \times_1 \mathbf{U}^c + \mathbf{L}^c \times_2 \mathbf{V}^c + \mathbf{F}^c \times_3 \mathbf{W}^c, \tag{5}$$

where c denotes class label, $\mathbf{H}, \mathbf{L}, \mathbf{F}$ are slice tensors that are composed of slice matrix components $\mathbf{H}_p, \mathbf{L}_q$ and \mathbf{F}_r respectively. Correspondingly, $\mathbf{U}, \mathbf{V}, \mathbf{W}$ are matrices composed of $\mathbf{u}_p, \mathbf{v}_q, \mathbf{w}_r, p = 1 \dots P, q = 1 \dots Q, r = 1 \dots R$ respectively.

Thus, the new EEG tensor data \mathbf{X} with unknown class label can be represented by giving class-specific slice tensors obtained from Eq.(5) and the coefficient matrices $\hat{\mathbf{U}}, \hat{\mathbf{V}}, \hat{\mathbf{W}}$ estimated by the SOD with fixed $\mathbf{H}, \mathbf{L}, \mathbf{F}$. However, SOD is based on iterative algorithm that starts from random matrices, which leads to the non-uniqueness problem. To further simplify this problem, we project \mathbf{X} on horizontal, lateral or frontal slice tensors separately by

$$\hat{\mathbf{U}} = \mathbf{X}_{(1)} \mathbf{H}_{(1)}^T (\mathbf{H}_{(1)} \mathbf{H}_{(1)}^T)^{-1}, \tag{6}$$

$$\hat{\mathbf{V}} = \mathbf{X}_{(2)} \mathbf{L}_{(2)}^T (\mathbf{L}_{(2)} \mathbf{L}_{(2)}^T)^{-1}, \tag{7}$$

$$\hat{\mathbf{W}} = \mathbf{X}_{(3)} \mathbf{F}_{(3)}^T (\mathbf{F}_{(3)} \mathbf{F}_{(3)}^T)^{-1}, \tag{8}$$

where $\mathbf{H}, \mathbf{L}, \mathbf{F}$ are class-specific slice tensors obtained by Eq.(5). Finally, we calculate the correlation coefficients of corresponding vectors between $\hat{\mathbf{U}}, \hat{\mathbf{V}}, \hat{\mathbf{W}}$ and $\mathbf{U}, \mathbf{V}, \mathbf{W}$ as features by

$$\mathbf{f} = [\mathbf{r}_{\hat{u}_p u_p}, \mathbf{r}_{\hat{v}_q v_q}, \mathbf{r}_{\hat{w}_r w_r}]_{p=1 \dots P, q=1 \dots Q, r=1 \dots R}. \quad (9)$$

For the classification of two classes EEG tensors, we first obtain class-specific slice tensors by applying SOD on the c -th class averaged tensor data $\underline{\mathbf{Y}}^c$. Thus, the \mathbf{f} for each of two classes are calculated by Eq.(6-9) with giving class-specific $\underline{\mathbf{H}}^c, \underline{\mathbf{L}}^c, \underline{\mathbf{F}}^c$ and combined into one feature vector that are used to train a linear classifier.

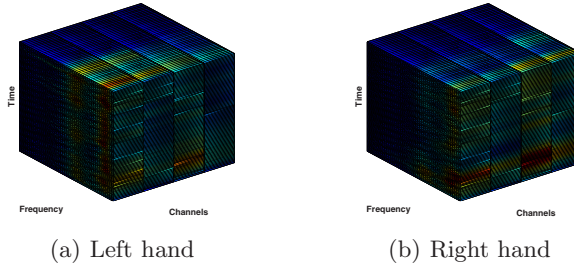


Fig. 2. Averaged 3-way tensors of space-frequency-time representation for EEG signals during MI tasks. The size of tensor data is $5 \times 49 \times 1024$ (i.e., channels \times frequency \times time). (a) for left hand class and (b) for right hand class.

3 Experiments and Results

In our application, EEG signals with only 5 electrodes (i.e., C3, Cp3, Cz, Cp4, C4) over the motor cortex were recorded from the scalp at a sampling rate of 256Hz for 2 classes MI-based BCI experiments. In the experimental sessions used for the present study, labeled trials of EEG signals were recorded in the following way: the subjects were sitting in a comfortable chair with arms lying relaxed on the armrests. Each trial consists of 2s for relaxation and 4s for movement imagination (i.e., left hand or right hand) tasks following visual cue stimulus.

The EEG data are transformed from the time-domain to the time-frequency domain using a complex Morlet continuous wavelet transform (CWT) with center frequency $\omega_c = 1$ and bandwidth parameter $\omega_b = 2$. The frequency range from 6Hz to 30Hz at 0.5Hz step are focused in our application. Thus, we obtain EEG tensor representation $\underline{\mathbf{X}} \in \mathbb{R}^{N_d \times N_f \times N_t}$ which is a 3-way time-varying EEG wavelet coefficients array, where N_d, N_f, N_t are the number of channels, frequency bins, and time points respectively. In our application, we only consider the time-frequency power features of EEG trials, hence a square operation is performed on $\underline{\mathbf{X}}$ in advance. In order to find the invariable feature structure through all trials, we first preprocessed EEG tensors by averaging the same class as $\underline{\mathbf{Y}}^c = \frac{1}{M} \sum_{i \in \text{class}_c} \underline{\mathbf{X}}_i$, M is the trial number of c -th class. Fig. 2 shows the 3D averaged tensors for each class.

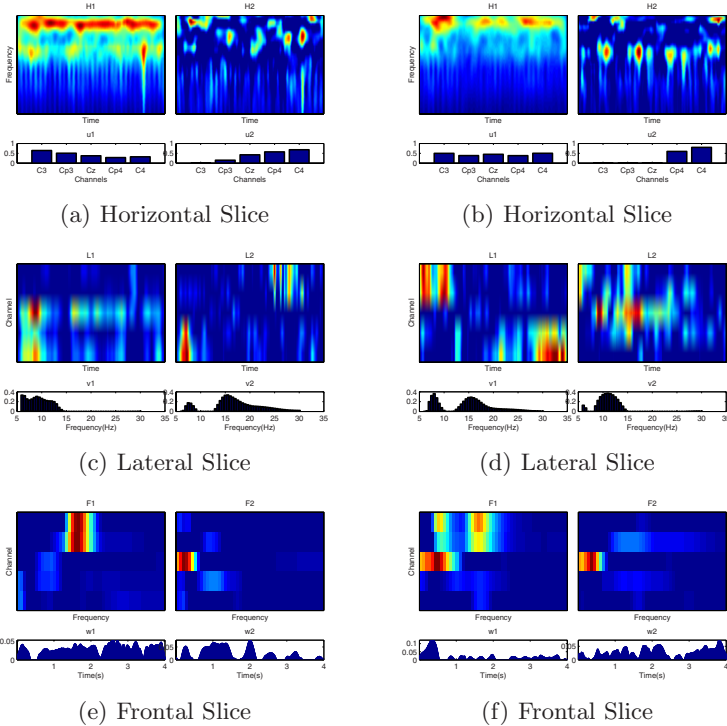


Fig. 3. Results of NN-SOD with $P = Q = R = 2$ applied to class-specific EEG tensors. (a)(c)(e) are slice components for left class, (b)(d)(f) are slice components for right class. (a)(b) are horizontal slice matrix \mathbf{H}_p and combination vector \mathbf{u}_p ; (c)(d) are Lateral slice matrix \mathbf{L}_q and combination vector \mathbf{v}_q ; (e)(f) are frontal slice matrix \mathbf{F}_r and combination vector \mathbf{w}_r . The tensor size is $5 \times 49 \times 1024$, i.e., 5 channels, 49 frequency bins and 1024 sample points.

The SOD model with non-negative constraints was performed for slice decomposition on each of class-specific $5 \times 49 \times 1024$ tensors (i.e., the space, frequency and time domain). In order to represent the tensor data by slice components with the number as smaller as possible, the fitting error of 0.1 is used for selection of components number. To simplify this procedure, we choose the same number for horizontal, lateral and frontal slice components. Fig. 3 presents the decomposition results with 2 components on each mode, i.e., $P = Q = R = 2$. In the horizontal slice components (Fig. 3(a)) for left class, the time-frequency matrix \mathbf{H}_1 mainly focuses around 10Hz (μ -rhythm) throughout the whole 4s duration of one trial. Then the vector \mathbf{u}_1 which represents the space distribution of the corresponding slice demonstrates that the slice \mathbf{H}_1 is decreased from channel C3 to C4. This is the ERS phenomena. Meanwhile the slice \mathbf{H}_2 and corresponding vector \mathbf{u}_2 demonstrate the ERD phenomena of decreasing power

of μ -rhythm on right motor area of brain. Similar to left class, Fig. 3(b) shows the time-frequency slice components and the distribution on channels domain, \mathbf{H}_1 denotes interrupt of μ -rhythm on left and right hemisphere of brain, \mathbf{H}_2 denotes low β -rhythm on left hemisphere of brain. Therefore, the significance of ERD/ERS for left hand and right hand are not same for specific subject. Similar to the horizontal slices, Fig. 3(c), 3(d) present the space-time lateral slice components and distribution vectors in the frequency domain. Fig. 3(e), 3(f) present the space-frequency frontal slice components and distribution vectors in the time domain. In Fig. 3(e), \mathbf{F}_1 denotes the β -rhythm focused on the left hemisphere of brain.

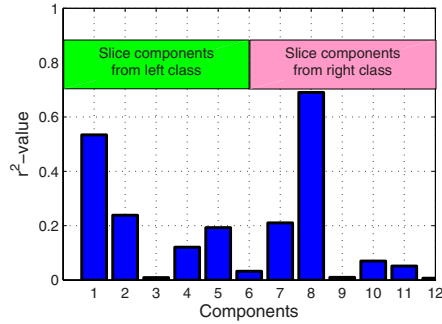


Fig. 4. r^2 -value of slice components. The first 6 components are obtained by left class tensor, and the last 6 components are obtained by right class tensor. The order of each 6 components are $\mathbf{H}_1, \mathbf{H}_2, \mathbf{L}_1, \mathbf{L}_2, \mathbf{F}_1, \mathbf{F}_2$.

In order to find the most discriminative slice components for two mental tasks, the r^2 -value are calculated for each slice components. Based on this, the 4 most discriminative slice components are selected for classification. Fig. 4 shows r^2 -value along slice components, the first 6 components for left class and the last 6 components for right class. It can be clearly seen that \mathbf{H}_1 of left class and \mathbf{H}_2 of right class have most discriminative ability, which illustrates that most discriminative information between left and right class lies in the space distribution of time-frequency slices. This just demonstrated why we can obtain high performance only by spatial filters, e.g., CSP method. However, the CSP algorithm is also known for its tendency to overfit, i.e., to learn the non-discriminative brain rhythm which has an overlapping frequency range with most discriminative brain rhythm. Especially in the small training samples case, CSP is suffered for the outlier problem because of high dependence upon the distribution properties of training data. As compared with CSP, our method are more stable in case of small training samples and more robust to deal with the nonstationary of EEG signals. To prove that, we has trained a SOD model and SVM classifier on first experiment run and tested our method on several subsequent

runs. Fig. 5 presents the classification performance of 10 runs for subject A and 8 runs for subject B. The results demonstrate that the relative stable performance can be obtained by our method when compared with CSP. Therefore, the generalization ability of our method seems to be more suitable for online BCI system.

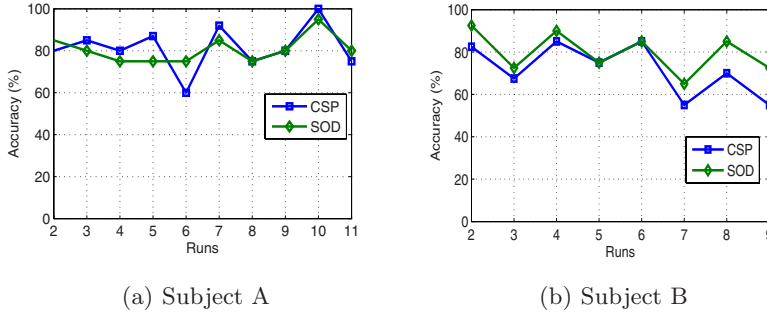


Fig. 5. The classification performance of subsequent runs based on the model trained on the first run. One experiment run contains only 20 trials for each class and the duration of mental tasks is 4s for each trial. (a) for subject A and (b) for subject B.

4 Conclusions

In this study, we have presented a novel tensor feature extraction framework for EEG classification based on SOD algorithm. Through applying the non-negative SOD, the slice features on each tensor mode can be easily obtained. Data analysis on EEG signals from BCI experiments demonstrates the effectiveness of our method. Compared with traditional tensor learning methods, our method is able to extract slice matrices from tensor data on multi-mode simultaneously, hence the space-frequency, space-time, and time-frequency structure features can be captured from 3D tensor data. Classification performance on several experiment runs also confirmed the robustness of our method. To further improve the discriminative ability, the class information will be additionally considered in the cost function and the semi-supervised feature extraction method will be studied in the next step.

Acknowledgments

The work was supported in part by the Science and Technology Commission of Shanghai Municipality (Grant No. 08511501701) and the National Natural Science Foundation of China (Grant No. 60775007).

References

1. Smilde, A.K., Bro, R., Geladi, P.: *Multi-way Analysis with Applications in the Chemical Sciences*. Wiley, Chichester (2004)
2. Heiler, M., Schnörr, C.: Controlling Sparseness in Non-negative Tensor Factorization. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3951, pp. 56–67. Springer, Heidelberg (2006)
3. Cichocki, A., Zdunek, R., Choi, S., Plemmons, R., Amari, S.: Non-Negative Tensor Factorization using Alpha and Beta Divergences. In: *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007*, vol. 3 (2007)
4. Mørup, M., Hansen, L., Herrmann, C., Parnas, J., Arnfred, S.: Parallel Factor Analysis as an exploratory tool for wavelet transformed event-related EEG. *Neuroimage* 29(3), 938–947 (2006)
5. Miwakeichi, F., Martínez-Montes, E., Valdés-Sosa, P., Nishiyama, N., Mizuhara, H., Yamaguchi, Y.: Decomposing EEG data into space–time–frequency components using Parallel Factor Analysis. *Neuroimage* 22(3), 1035–1045 (2004)
6. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.: *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley, Chichester (2009)
7. Mørup, M., Hansen, L., Arnfred, S.: ERPWAVELAB A toolbox for multi-channel analysis of time–frequency transformed event related potentials. *Journal of Neuroscience Methods* 161(2), 361–368 (2007)
8. Yang, J., Zhang, D., Frangi, A., Yang, J.Y.: Two-dimensional pca: A new approach to appearance-based face representation and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 26(1), 131–137 (2004)
9. Tao, D., Li, X., Wu, X., Maybank, S.: General Tensor Discriminant Analysis and Gabor Features for Gait Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(10), 1700–1715 (2007)
10. Ye, J., Janardan, R., Li, Q.: Two-dimensional linear discriminant analysis. In: *NIPS* (2004)
11. Wang, X., Tang, X.: A unified framework for subspace face recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* 26(9), 1222–1228 (2004)
12. Fu, Y., Huang, T.: Image classification using correlation tensor analysis. *IEEE Transaction on Image Processing* 17(2), 226–234 (2008)
13. He, X., Cai, D., Niyogi, P.: Tensor subspace analysis. In: *NIPS 2006* (2006)
14. Tao, D., Li, X., Hu, W., Maybank, S., Wu, X.: Supervised tensor learning. In: *Proceedings of the Fifth IEEE International Conference on Data Mining, ICDM 2005* (2007)
15. Caiafa, C.F., Cichocki, A.: Slice Oriented Decomposition (SOD): A New Tensor Decomposition for Representation of 3-way Data (submitted February 2009)

Stereo Map Surface Calculus Optimization Using Radial Basis Functions Neural Network Interpolation

Allan David Garcia de Araujo¹, Adriaio Duarte Doria Neto¹,
and Allan de Medeiros Martins²

¹ Departamento de Engenharia de Computacao e Automacao
Universidade Federal do Rio Grande do Norte
Natal, Rio Grande do Norte, Brasil

² Departamento de Engenharia Eletrica
Universidade Federal do Rio Grande do Norte
Natal, Rio Grande do Norte, Brasil

allan.garcia@gmail.com, adriao@dca.ufrn.br, allan@dee.ufrn.br
<http://www.ppgeec.ufrn.br>

Abstract. Matching two points in distinguished images is one of the actual challenges in digital image processing. The currents techniques of stereo imaging are not ideal and slow, not offering a valuable solution for practical scenarios like real time robotics vision or vehicles navigation. This paper will presents a approach for optimization of disparity calculus, introducing a sparse matching of stereo disparity maps and surface reconstruction by RBFs interpolation of empty spaces.

Keywords: Neural Networks, RBF, Computer Vision, Stereo Matching.

1 Introduction

The acquisition of an image on a modern digital camera records the light perception that reaches the internal sensors of that camera. This acquisition is a projection of a three-dimensional scene on a two-dimensional plan, which is the lens of the camera. Remember that the mapping of a three-dimensional scene on a image plane is a “many-to-one” transformation, ie, one point on the picture doesn’t determines a single position in the world [1], what makes necessarily some techniques to make this information acessible somehow.

In applications such as: robotic navigation, geographic mapping, et al., Where the depth information (Z coordinate) of scene points are often necessary, we must use a technique known as stereoscopic imaging (or stereo) to recover the lost information.

The stereoscopy, or stereo vision, is the particular case of computer vision processing that uses a basics two-dimensional images, acquired by a system of two or more cameras, to estimate the dimension of depth on a scene [6]. In this case, each device sees the scene from two different references, allowing the

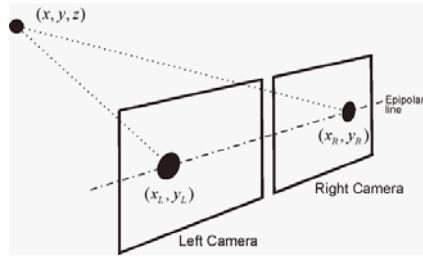


Fig. 1. A common scenario of stereo vision

computer system to combine the information obtained from the pair of images in order to obtain a three-dimensional representation. This model has strong biological inspiration and is the way of our human vision works.

The combination of two or more two-dimensional images, which aims to locate the corresponding points between an image and another isn't a trivial task. It is required adjustments of parameters that are dependent of the scene and the computational time is extremely high. Two main techniques commonly used to find this correlation are based on the technical areas [7][8] and techniques based on characteristics [9].

The techniques based in areas uses the correlation between the values of intensity of a window on the left image, and another on the right image, producing a dense map of disparities. The size of window on this algorithm and the search area, influences the accuracy of the match and also the complexity of processing [10].

The techniques based on features uses the characteristics removed from the images, such as segments of the edge, gradients, etc., performing simple comparisons between attributes these characteristics. Such techniques have enhanced speed and accuracy than the techniques based in area, however, have the disadvantage of generating a disparity map of sparse, being necessary to make use of interpolation techniques to generate dense maps [5].

Besides these two techniques, it shows, as a powerful alternate, the techniques based on Artificial Neural Networks to solve the problem of correspondence. In this field of work we can cite the work of Hsiao and Wang [4], which makes use of MLPs with supervised training to find the points correspondents.

The problem of stereo vision matching is extremely complex and not always provide definitive solutions. Many are the difficulties of such processing, which among others can quote geometric distortions, radiometric, regions of occlusions and similarities between points [11]. Another dominant factor is the high computational cost of current techniques.

This work aims to combine the simplicity of technique of area based stereo matching using color images, but without making an ostensible calculation for all points of the image, thus making the algorithm very slow. To fill the gaps left by the algorithm of matching, take a rather consolidated technic of points interpolation using Artificial Neural Networks with Radial Base Function (RBFs)

in order to find the other points of the surface that were not calculated. We thus reconstruct the map of the depths of the original scene with some gain in performance and with minimum possible error.

This paper is organized as follows: First, set the basic concepts to understanding the techniques used, then talk on the proposed method, its characteristics and propositions, then present the results and, finally, finished with the conclusions and proposals for future work.

2 Stereo Vision

The requirement for stereo vision is the existence of two separate views of a scene of an object.

For the proposed scenario we assume that the cameras are identical and that coordinate systems of both cameras are perfectly aligned, differing only in the position of their origins. We noticed that the camera and world coordinate systems coincide, the plan xy of the image is aligned with the plane (X, Y) of the world coordinate system. So, in these conditions, the coordinate Z of \mathbf{w} is exactly the same for both coordinate systems of cameras.

2.1 Stereo Images Correlation

The correlation can be used as a matrix for finding a matching subimage $w(x, y)$ with size of $J \times K$ inside an image $f(x, y)$ of size $M \times N$, assuming that $J \leq M$ and $K \leq N$.

The correlation between $f(x, y)$ and $w(x, y)$ can be expressed by:

$$c(s, t) = \sum_x \sum_y f(x, y)w(x - s, y - t)$$

where $s = 0, 1, 2, \dots, M - 1$ and $t = 0, 1, 2, \dots, N - 1$, and the sum is realized over a region of the image with f and w are placed.

We assume that the origin of $f(x, y)$ is the top left and for $w(x, y)$ in its center. As s and t are scanned, the $w(x, y)$ are moved in the image, providing a function $c(s, t)$. The maximum value of $c(s, t)$ indicates the position in which $x(x, y)$ has better matching with $f(x, y)$.

3 Artificial Neural Networks

The architecture of an Artificial Neural Network can be approached in several ways. As an example we can cite the case of the *back-propagation* algorithm that can be used to train MLP supervised networks (networks with multiple layers of perceptrons), which is applied to stochastic approximation of a function, however, when the dimensionality of the curve are very high, the algorithms convergence are extremely slow.

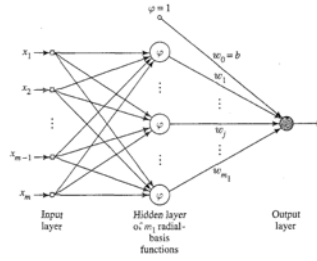


Fig. 2. RBF schematics

In contrast, networks of radial basis functions (RBFs), illustrated above, [12], can be seen as an approach of *curve-fitting*, or approximation, but with high-dimensional data [2]. In this context, the learning process consists to create an n-dimensional surface that offers the best fit to the training data, which the stop criterion is a statistical measure. Similarly, the generalization is equivalent to creating a hyper-surface to interpolate the training data. Its architecture has a hidden layer of neurons with a characteristic different from traditional MLP, which is exactly the presence of the radial basic functions in the entry standards of data used for training.

Thus, we can say that the RBF networks are suitable to interpolate large masses of data with extreme efficiency.

4 Proposed Method

Our work has developed a technique to oversee the complexity of classical stereo matching algorithms [13], and to improve the performance of an area stereo matching, a short explanation can be follow on Fig. 3.



Fig. 3. Block diagram of proposed method

The approach adopted was to reduce the number of points calculated introducing in the algorithm a jump parameter (*step*), in pixels, that is the intervals that will be considered in the calculus of matching pixels. The result is the creation of an imaginary grid of spacing where only in crosses of the lines is that the differences are calculated.

After the imaginary mesh has been created, the processing of stereo matching is started. For purposes of matching, we consider a point \mathbf{w} into (x_1, y_1) on the right image is similar to \mathbf{v} at (x_2, y_2) from the left image, by the correlation of the vector formed by points included in a window of size J_1 for a vector of the

search space of S in the image on the right is the highest possible. It is therefore, that the pair $P(w, v)$ are related. Note that due to the parameter of the jump, the result here will be a sparse map the gaps, and needs to be interpolated.

After this stage, we began the neural interpolator using the technique of RBF to reconstruct the lost area, so we choose to put all coordinates of map of the calculated differences as a centers of the neural network.

After training the network we create a second mesh of points with resolution equal to the original image. This mesh is used to simulate the outcome data of interpolated data calculated by RBF.

5 Results

For comparison, we tested running a traditional area based stereo matching algorithm on a set of images, this algorithm creates disparity maps, without interpolation, using various values for *window* and *search* parameters. We randomly selected two images from this set to use on this paper, as we can see on Fig. 4 and Fig. 5. These images will reference to what would be an ideal state.



Fig. 4. Example 1 disparity map without the interpolation with stereo parameters 15px for window and 50px for search



Fig. 5. Example 2 disparity map without the interpolation with stereo parameters 18px for window and 30px for search

We're calling here *window* a collection of pixels that are in the region of the central pixel, this collection describes the central pixel itself in this unique position. And the *search* parameter describes the range of seek that we will search for a pixel in the right image for a match pixel of the one on the left image. Those two parameters are very important to get a good result, and those two values were choosed after severals experiments tests.

This map of disparities were found through the matching of the original images below in Fig. 6 and Fig. 7.

Our experiment consists in, using the proposed method, significantly increase the *step*, in the way that, for each iteration, less data is calculated from the stereo pair, being necessary that the RBF interpolates an increasing range of pixels, until we notice that the interpolated image no longer retains the characteristics of the original disparity map.



Fig. 6. Left image of stereo pair for Example 1 disparity map



Fig. 7. Left image of stereo pair for Example 2 disparity map

The RBF has been calibrated to work efficiently for our objective, which is, to interpolate data that are similar in a three-dimensional geometry. The parameters used were chosen based on the best responses after a wide range of tests. The geometric values of X and Y are loaded using the size of the resulting mesh of sparse stereo map, the **RBFFunction** are the *multiquadric* function, for better results on edges areas, the **RBFConstant** are the expression $y = prod(max(x') - min(x'))/nXCCount)^{(1/nXDim)}$ which means the approximated average distance between the nodes, and the **RBFSmooth** is set to zero.

We run our method for each image using the values of *step* from 1 to 100 pixels, but for visualization we'll show three states for the two examples images, which are with step 10, step 15 and with step 50, respectively.

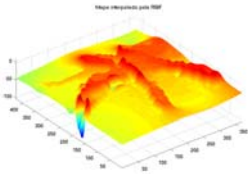


Fig. 8. Surface rebuilt by RBF of Example 1 using a *step* of 10 pixels

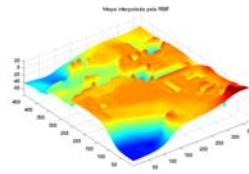


Fig. 9. Surface rebuilt by RBF of Example 2 using a *step* of 10 pixels

Note that after a certain amount of *steps*, as expected, the surface no longer has the characteristics of the real disparity information. This is because too much interpolated data are created and the calculated stereo map no longer has the characteristics of the original map of disparities.

Comparing the interpolated data with the initial stereo map of disparities, we'll get the error differences of this method, and as result we obtained the graphs below on Fig. 14 for Example 1 and Fig. 15. They shows two curves each of analyzed data on experiments. The x-axis are the steps, and the y-axis the values of the time and the error. The solid curve represents how the CPU time are decreased as the steps are increased, and the dashed curve is the error.

We can see the the results are similar for diferents values of *window* and *search*, and for diferents images.

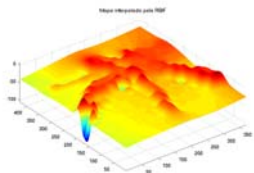


Fig. 10. Surface rebuilded by RBF of Example 1 using a *step* of 15 pixels

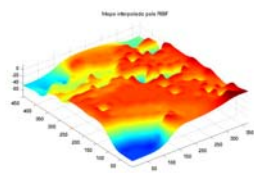


Fig. 11. Surface rebuilded by RBF of Example 2 using a *step* of 15 pixels

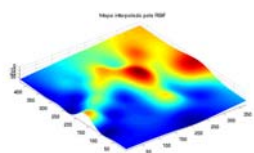


Fig. 12. Surface rebuilded by RBF of Example 1 using a *step* of 50 pixels

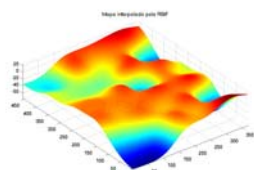


Fig. 13. Surface rebuilded by RBF of Example 2 using a *step* of 50 pixels

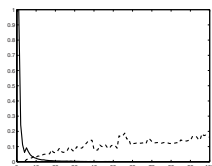


Fig. 14. The Error X Time graph for Example 1

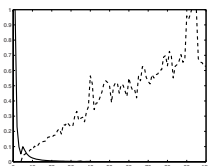


Fig. 15. The Error X Time graph for Example 2

The same data in tabular form for Example 1:

Table 1. The comparison table between step and interpolation error

STEP (pixels)	TOTAL TIME (sec)	ERROR
1	1999,59	0
10	58,46	0,03
30	18,99	0,07
100	17,66	0,16

6 Conclusions

Solving the problem of stereo matching is extremely computationally expensive, but with some approaches we can reduce significantly the CPU time required to produce a near optimal maps of depth informations.

We realize that with a small jump in the calculation of the disparities, we gain significant performance with almost no error, and that the RBF networks are extremely useful in bringing the original surface.

References

1. Gonzalez, R.C., Woods, R.E.: *Processamento de Imagens Digitais*. In: Blcher, E, ed. (1992)
2. Haykin, S.: *Neural Networks, a comprehensive foundation*, 2nd edn. Pearson Education, London (2001)
3. Calin, G., Roda, V.O.: Real-time disparity map extraction in a dual head stereo vision system. *Lat. Am. Appl. Res. Mar.* 37(1), 21–24 (2007)
4. Wang, J.H., Hsiao, C.P.: On Disparity Matching in Stereo Vision via a Neural Networks Framework. *Natl. Sci. Counc. ROC* (1999)
5. Fernandes, R.G., Silveira, R.W., Dria Neto, A.D.: On Disparity Matching in Stereo Vision via a Neural Networks Framework. *Departamento de Engenharia de Computao e Automao - UFRN* (2004)
6. Marr, D.: *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., New York (1982)
7. Marapane, S.B., Trivedi, M.M.: Region-based stereo analysis for robotic applications. *IEEE Transactions on Systems, Man and Cybernetics* 19(6), 1447–1464 (1989)
8. Li, G., He, Y.: A hierarchical combined feature and area-based stereo matching algorithm. In: *IEEE International Symposium on Circuits and Systems, ISCAS 2002*, vol. 2, pp. II-277–II-280 (2002)
9. Goulermas, J.Y., Liatsis, P.: Feature Based Stereo Matching via Coevolution of Epipolar Subproblems. In: *Seventh International Conference on Image Processing And Its Applications*, vol. 1 (1999)
10. Sunyoto, H., Mark, W.V.D., Gavrilu, D.M.: A comparative study of fast dense stereo vision algorithms. In: *2004 IEEE Intelligent Vehicles Symposium*, June 2004, pp. 319–324 (2004)
11. Lin, J.H., Parhi, K.K.: VLSI architectures for stereoscopic video disparity matching and object extraction. In: *IEEE International Symposium on Circuits and Systems, ISCAS 2005*, May 2005, vol. 3, pp. 2373–2376 (2005)
12. Rahmatulloh, F.: *Radial Basis Function Networks for Modeling Option* (last seen June 2009), <http://statistikawan.org/radial-basis-function-networks-for-modeling-option.html>
13. Zhang, W., Zhang, Q., Qu, L., Wei, S.: A stereo matching algorithm based on multiresolution and epipolar constraint. In: *Proceedings of Third International Conference on Image and Graphics*, 2004, December 18–20, 2004, pp. 180–183 (2004)

Quasi-Deterministic Partially Observable Markov Decision Processes

Camille Besse and Brahim Chaib-draa

Department of Computer Science, Laval University, Quebec, Canada
{besse, chaib}@damas.ift.ulaval.ca

Abstract. We study a subclass of POMDPs, called quasi-deterministic POMDPs (QDET-POMDPs), characterized by deterministic actions and stochastic observations. While this framework does not model the same general problems as POMDPs, they still capture a number of interesting and challenging problems and, in some cases, have interesting properties. By studying the observability available in this subclass, we show that QDET-POMDPs may fall many steps in the complexity classes of polynomial hierarchy.

1 Introduction

AI planning was initially conceived as a deterministic problem where a sequence of actions has to be decided in order to achieve a goal state with desirable values from an original state. This problem was thoroughly studied in AI with important contributions as A^* , GRAPHPLAN, and others [1].

However, this deterministic model has strong limitations on the type of problem that can be represented. Thus, one cannot represent situations where actions have non-deterministic outcomes or where states are not completely observable. In such cases, one must resort to Markov Decisions Processes (MDPs) and Partially Observable Markov Decisions Processes (POMDPs). However, with this expressiveness of the model comes an increase of complexity, specially for POMDPs, and this gain in generality involves a cost in the ability to solve the sustained problems by such model. For instance, POMDPs offer one of the most expressive frameworks and are thus widely used for sequential decision making under partial observability [2], but the current known algorithms scales very poorly as the planning horizon grows.

Nevertheless, numbers of problems that involve partial observability have a common characteristic: they have actions with deterministic outcomes and the observation generated is also deterministic. Indeed, these problems have recently been used in many proposals for planning with incomplete information, e.g. [3], and are used for learning partially observable models [4].

These models were briefly discussed in [5], under the name of deterministic POMDPs (DET-POMDPs) for which some important theoretical results were obtained. Littman [5] first showed that a DET-POMDP can be mapped into an MDP with an exponential number of states and then be solved with standard algorithms for MDPs. He also showed that optimal non-stationary policies of polynomial size can be computed in non-deterministic polynomial time and finally that optimal stationary policies can be computed in polynomial space. Since then, up to our knowledge, no publications were made on this

subject except [6] that extends these results by defining a specific subclass of DET-POMDPs, that have the so-called *polynomial diameter* property, that can be solved in non-deterministic polynomial time. Bonnet [6] also linked the DET-POMDP framework to the AND/OR tree search algorithms, arguing that this type of algorithm is more efficient than standard POMDP algorithms for this subclass of POMDPs.

Given this role of DET-POMDPs in recent research and motivated by the quest of amenable models for decision making under partial observability, we extend the work of Littman and Bonnet in order to bridge a part of the gap between DET-POMDPs and POMDPs, by studying the subclass of POMDPs with deterministic transition but with stochastic observations. We thus present a specific subclass of widely used DET-POMDPs, called quasi-DET-POMDPs (QDET-POMDPs) and show that ε -approximating this subclass falls many steps in complexity in the polynomial hierarchy.

This paper is organized as follows. First, examples of challenging problems are given that motivate our research. Second, a formal definition of the model and the variants are given. In Sect. 4 main theoretical results are described and the complexity of the subclass is presented. Finally, the significance of this work is discussed in Sect. 5.

2 Examples

Many problems had been modeled as POMDPs and DET-POMDPs and had been used to develop and evaluate various algorithms for planning under uncertainty and partial information. For space reasons, we present only few examples of some problems that may be modeled as a QDET-POMDP:

Robot Navigation: Consider an indoor robot in a $m \times n$ grid that must navigate from an initial position to a goal position while avoiding obstacles using only some noisy sensors on its position. The robot's moves are deterministic but the observation of its current state is distorted by the noise on the sensors. The goal is to find a strategy for guiding the robot to its destination.

Diagnosis: The aim of diagnosis is to identify one of the m states of a system (e.g. a patient) using n noisy binary tests. An instance consists of a $m \times n$ stochastic matrix T where each T_{ij} represent the probability that test j is positive in the state i . The goal is to find the sequence of tests that will identify almost surely the state of the studied system [7].

Sensor Management: Consider multiple sensors situated on a single platform where each sensor can be activated solely (e.g. Figure 1). The problem is to track a concealed or distant target by interrogating the sensors. The target is modeled by a set of states, each state representing a contiguous set of target-sensor orientations over which the scattering physics is relatively stationary. The goal is to find a tracking policy for the target while observing only noisy relative sensor angular positions [8].

All of these problems can be modeled as QDET-POMDPs. Let us now see the formal definition of the proposed framework.

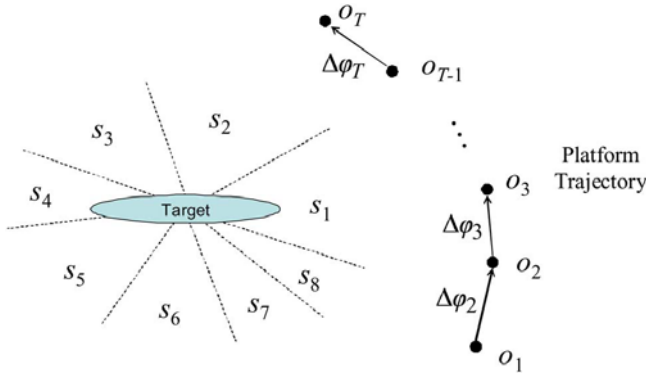


Fig. 1. Multi-aspect sensing of a hidden target. The k th state s_k is a contiguous set of target-sensor orientations over which the scattered fields are approximately stationary ($K = 8$ states are shown). Here T observations are performed, $\{o_1, o_2, \dots, o_T\}$, as performed at a sequence of relative sensor angular positions, where $\Delta\varphi_{t+1} = \varphi_{t+1} - \varphi_t$ are orientations. Figure from [8].

3 Model and Variants

Deterministic POMDPs were initially defined as follows [5]:

Definition 1. [5] A *Deterministic Partially Observable Markov Decision Process* (DET-POMDP) is a tuple $\langle \mathcal{S}, \mathcal{A}, \Omega, \mathcal{T}, \mathcal{O}, \mathcal{R}, \gamma, \mathbf{b}^0 \rangle$, where:

- \mathcal{S} is a finite set of states $s \in \mathcal{S}$;
- \mathcal{A} is the finite set of actions of the agent and $a \in \mathcal{A}$, denotes an action;
- Ω is the finite set of observations of the agent and $z \in \Omega$, denotes an observation;
- $\mathcal{O}(z, a, s') : \Omega \times \mathcal{A} \times \mathcal{S} \mapsto \{0, 1\}$ is the deterministic observation function indicating whether or not the agent gets observation z when the world falls in state s' after executing action a ;
- $\mathcal{T}(s, a, s') : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \{0, 1\}$ is the deterministic transition function indicating whether or not making action a in state s results in state s' ;
- $\mathcal{R}(s, a) : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward perceived by the agent when the world falls into state s after agent executes action a ;
- γ is the discount factor;
- \mathbf{b}^0 is the a priori knowledge about the state, i.e. the initial belief state, assumed non-deterministic.

The variant proposed by [6] considers a set of absorbing goal states that provide no rewards nor costs but is semantically similar.

Note that the initial belief state \mathbf{b}^0 , which describes the different possibilities for the initial state, is crucial. Indeed, if the initial state were known, and since the transition function is deterministic, then all the future states will also be known, and the model reduces to the well studied problem of deterministic planning in AI [1].

Compared to deterministic POMDPs, the proposed extended model presents changes on the observability function. This model, so called Quasi-deterministic Partially Observable Markov Decision Process, is defined as follows:

Definition 2. A *Quasi-deterministic Partially Observable Markov Decision Process* (QDET-POMDP) is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{O}, \mathcal{R}, \mathbf{b}^0 \rangle$, where:

- $\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathbf{b}^0$ are the same as in Definition 1;
- $\mathcal{O}(z, a, s') : \Omega \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ is the observation function indicating the probability of getting observation z when the world falls in s' after executing a ;
 Moreover, $\forall s' \in \mathcal{S}, a \in \mathcal{A}, \exists z \in \Omega, s.t. \mathcal{O}(z, a, s') \geq \theta > \frac{1}{2}$, i.e. the world is minimally observable and the probability of getting one of the observations is lower bounded in each state by at least one half;

First, let us notice that θ is just a lower bound on observability of each state and thus that in some states the probability the observation can be greater. Notice also that the planning horizon is not set *a priori*. This is due – as we will see in Section 4 – to an interesting convergence property of this model to a ε -deterministic belief state after a fixed number of steps.

Optimality Criteria and Variants

As our goal is to compute a policy that permits our agent to perform *optimally*, we consider the **maxexp** optimality criteria that maximizes the expected discounted reward of a policy. The value of a policy π is thus computed using:

$$V_\pi(\mathbf{b}^0) = \mathbb{E}_{s \sim \mathbf{b}^0} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s^t, \pi(s^t)) \mid s^0 = s, \pi \right]$$

The variants of the model are related to the observation model:

Unobservable models in which $|\Omega| = 1$ and thus no information is retrieved about the state. This class is a subclass of the so-called conformant problem in planning [9].

Fully Observable models in which $\Omega = \mathcal{S}$ and $\mathcal{O}(z, a, s') = 1$ iff $z = s'$. This class is exactly the classic fully observable MDPs where only the initial state is unknown.

Non-unobservable models in which $|\Omega| > 1$. This class is exactly the complement of unobservable problems. Among this class of problems, we distinguish:

Enough-observable models in which $\Omega = \mathcal{S}$. This class regroups all the linear but noisy observation problems where the state itself is perceived but with an additive noise. This class regroups for example all control problems where the state is perceived through noisy sensors.

Factored-observable models in which $|\Omega| = |\mathcal{X}| \times |\mathcal{D}_x|$. Where \mathcal{X} is the set of state variables and \mathcal{D}_x is the domain of variable x . The state space is then given by $\mathcal{S} = \prod_{x \in \mathcal{X}} \mathcal{D}_x$. This class is similar to the previous one using additive noise but restrain the number of observations along the “dimensions” of the state space. Indeed, as the state space is assumed structured, the agent can use this structure to learn about at least one dimension at each time step. The previous class of models is a restriction of this class with only one dimension.

General models which includes previous cases, does not assume anything on the observation function.

As the fully observable, the unobservable and the general cases were extensively studied in the literature [32], we will not consider them in the remaining of the paper. However, the enough-observable and the factored-observable cases present an interesting avenue since many of the quasi-deterministic problems mentioned earlier are very often factored or at least enough-observable. We will show in the next section that these problems actually are easier than the general problems by bounding the history needed to identify almost surely the underlying state.

4 QDET-POMDP Theoretical Analysis

In this section, a lower bound on the number of steps to ensure convergence to a certain belief is given and induced complexity results are explained.

As mentioned earlier in the paper, a way to represent compactly the full history of observations during the planning process is the *belief state* [10]. This is a probability distribution over the states that represents the belief of the agent to be in each state through probabilities. We denote by $\mathbf{b}^t(s) = \Pr(s|z^t, a^t, \mathbf{b}^{t-1})$ the probability of being in state s at step t given that observation z^t was perceived and action a^t was performed in the belief state \mathbf{b}^{t-1} . This probability is computed using Bayes' rule:

$$\mathbf{b}^t(s) = \frac{\mathcal{O}(z^t, a^t, s) \sum_{s' \in \mathcal{S}} \mathcal{T}(s', a^t, s) \mathbf{b}^{t-1}(s')}{\sum_{s'' \in \mathcal{S}} \mathcal{O}(z^t, a^t, s'') \sum_{s' \in \mathcal{S}} \mathcal{T}(s', a^t, s'') \mathbf{b}^{t-1}(s')} \quad (1)$$

Using a matrix representation, Equation (1) can be rewritten:

$$\mathbf{b}^k(s) = \frac{D_k T_{a^k} \cdots D_1 T_{a^1} \mathbf{b}^0}{\mathbb{1}^\top D_k T_{a^k} \cdots D_1 T_{a^1} \mathbf{b}^0} \quad (2)$$

Where \mathbf{b}^0 is the initial belief, T_{a^t} are transition matrices according to action a^t , D_i are diagonal matrices with the terms on the diagonal corresponding to the probability to observe z_i given each state, and $\mathbb{1}$ a $|\mathcal{S}|$ -dimensional vector of ones.

In order to show the convergence of the belief state to a single state with high probability, let us first state that this probability depends on the number n of succeeded observations among k steps in a non-unobservable context. Nevertheless, non-unobservability is not a sufficient condition to ensure this convergence. Let us now study how n varies regarding to the proposed variants on the observability.

4.1 Enough-Observable Models

Enough-observable models ensure that there is only one most likely observation (MLO) in each state and that each state's MLO is not the MLO of any other state:

Definition 3. *An enough-observable QDET-POMDP is a QDET-POMDP where following assumption holds:*

$$\begin{aligned} &\exists o_1 \in \Omega, \forall a \in \mathcal{A}, \forall s \in \mathcal{S}^{o_1}, \\ &\mathcal{S}^{o_1} = \{s \in \mathcal{S}, o_1 \in \Omega | P(o_1|s, a) > P(o|s, a), \forall o \neq o_1\}, \\ &|\Omega| = |\mathcal{S}| \text{ and } |\mathcal{S}^{o_1}| = 1 \end{aligned}$$

Here, \mathcal{S}^{o_1} is the set of states where o_1 is the MLO.

Considering this definition, one can state our first main result:

Theorem 1. *Under the enough-observability assumption, $\mathbf{b}^k(s) \geq 1 - \varepsilon$ iff*

$$n \geq \frac{1}{2 \ln \frac{\nu\theta}{(1-\theta)}} \ln \left[\frac{1 - \varepsilon}{\varepsilon} \left(1 + \nu^{1 - \frac{k}{2}} \right) \right] + \frac{k}{2} \tag{3}$$

Where $\nu = \max_{s,a} \sum_{z \in \Omega} I(\theta > \mathcal{O}(z, a, s) > 0) < |\Omega|$ the maximum number of “bad” observations that can be perceived in a state.

Proof (Sketch). In the worst case, the probability of observing the real underlying state is always minimal and equals to θ at each step. Moreover, if the failed observations obtained always support the second most likely state, it results in an increasing of the probability to potentially be in this state. According to Equation (2) and using determinism of transitions, which induces that transition matrices are permutation matrices, one must show that:

$$\frac{\theta^n \frac{(1-\theta)^p}{\nu^p}}{\theta^n \frac{(1-\theta)^p}{\nu^p} + \theta^p \frac{(1-\theta)^n}{\nu^n} + (\nu - 1) \frac{(1-\theta)^k}{\nu^k}} \geq 1 - \varepsilon \tag{4}$$

Where n is the number of successful observations of the real underlying state and $p = k - n$ the number of failures. The numerator is obtained by obtaining n times a “good” observation and p times a “bad” one during the execution. The denominator sum over all states the same sequence of observation where the first term is for the most likely state, the second term for the second most likely state and the third term for the rest of possible states according to the number of “bad” observations ν . We assume here that the probability to get a “bad” observation is uniform. This assumption is justified by the *maximum-entropy principle* which states that according to the current knowledge, the highest entropy distribution – the uniform in our case – is the best one. Solving¹ this inequality leads to Equation (3). □

Roughly speaking, ν represents also the way the error spreads over the false states.

4.2 Factored Models

In a more general way than enough-observable models, factored-observable models ensure that each value of each variable is sufficiently often observed so that the factored state can be determined in a finite number of steps:

Definition 4. *A factored-observable QDET-POMDP is a QDET-POMDP where following assumption holds:*

- The state space is factored in μ state variables: $\mathcal{S} = \times_{x \in \mathcal{X}} \mathcal{D}_x$ and observations possible are $\Omega = \bigcup_{x \in \mathcal{X}} \mathcal{D}_x$.
- The sum of probabilities of observing one state’s variables’ real values is lower bounded by $\theta > \frac{1}{2}$.

¹ An extensive derivation of the equations is given in Appendix A.

This definition implies that, in the worst case, for each state variable, there is a probability $\frac{\theta}{\mu}$ to observe its real value and a probability $\frac{1-\theta}{|\Omega|-\mu}$ to observe anything else. Note also that this definition is a generalization of Definition 3 which is the case $\mu = 1$. This statement leads to the following theorem:

Theorem 2. *Under the factored-observability assumption, $\mathbf{b}^k(s) \geq 1 - \varepsilon$ iff*

$$n \geq \frac{1}{2 \ln \frac{(|\Omega|-\mu)\theta}{\mu(1-\theta)}} \ln \left[\frac{1-\varepsilon}{\varepsilon} (1 + |\mathcal{S}| - \mu) \right] + \frac{k}{2} \tag{5}$$

Proof (Sketch). The proof follows exactly the same arguments as in Theorem 1. □

Once the number n of most likely observation is lower bounded, finding the probability to achieve at least this number is simply an application of the binomial distribution to have at least n successes on k trials:

Corollary 1. *In any QDET-POMDP and under Theorem 1 or Theorem 2 assumptions, the probability that a belief state $\mathbf{b}^k(s)$ is ε -deterministic after k steps is:*

$$\exists s, \Pr(\mathbf{b}^k(s) \geq 1 - \varepsilon) = \sum_{i=n}^k \binom{k}{i} \theta^i (1 - \theta)^{k-i} \tag{6}$$

Table 1. Enough-Observable bound

θ	ν	k	$n \geq$
0.6	3	75	40
0.6	10	59	31
0.6	100	50	26
0.7	3	22	13
0.7	10	19	11
0.7	100	14	8
0.8	3	9	6
0.8	10	6	4
0.8	100	6	4

Table 2. Factored-observable bound

θ	μ	$ \mathcal{D} $	$ \mathcal{S} $	k	$n \geq$
0.6	2	10	100	84	44
0.6	3	5	125	98	52
0.6	6	10	10^6	112	60
0.7	2	10	100	30	17
0.7	3	5	125	33	19
0.7	6	10	10^6	39	23
0.8	2	10	100	13	8
0.8	3	5	125	16	10
0.8	6	10	10^6	20	13

4.3 Experimentations

To give an idea of the efficiency of the proposed PAC bound, we define $\delta > 0$ such that $\Pr(\mathbf{b}^K(s) \geq 1 - \varepsilon) \geq 1 - \delta$. Table 1 and 2 give, for $\varepsilon = 10^{-3}$, $\delta = 10^{-1}$ and different values of θ , ν , μ and the domains' size of variables, the value of the bound on the horizon k and the number of successes needed n given that the probability of having both is above $1 - \delta$. As expected, horizons needed to converge are greater in the factored case than in the enough-observable case for similar state and observation spaces since the agent, at each time step, get less information about the current state. Actually, observations discriminate among subsets of states but not among states themselves like in the previous case. However, as the number of observations is much less than the previous case, current algorithms may have much less difficulty in this type of problems. An empirical study of their difference should be interesting as a research avenue. Let us now derive the worst case complexity from these bounds.

4.4 Impact on Complexity

A major implication of Theorems 1 and 2 is the reduction of the complexity of general POMDPs problems when a QDET-POMDP is encountered. Indeed, [11] have shown that finite-horizon POMDPs are PSPACE-complete. However, fixing the horizon T to be constant, causes to complexity to fall down many steps in the polynomial hierarchy [12]. In the case of constant horizon POMDP, one can state:

Proposition 1. *Finding a policy for a finite-horizon- k POMDP, that leads to an expected reward at least C is Σ_{2k-1}^P .*

Proof. To show that the problem is in Σ_{2k-1}^P , the following algorithm using a Σ_{2k-2}^P oracle can be used: guess a policy for $k-1$ steps with the oracle and then verify that this policy leads to an expected reward at least C in polynomial time by verifying the $|\Omega|^k$ possible histories, since k is a constant. \square

As QDET-POMDPs are a subclass of POMDPs and since fixing δ induces a constant horizon under Theorem 1 or Theorem 2 assumptions:

Corollary 2. *Finding a policy for a QDET-POMDP, under Theorem 1 or 2 assumptions, that leads to an expected reward at least C with probability $1-\delta$, is Σ_{2k-1}^P .*

Practically, finding a probably approximatively correct ε -optimal policy for a QDET-POMDP thus implies using a k -QMDP algorithm that computes exactly k exact backups of a POMDP and that then uses the policy of the underlying MDP for the remaining steps (eventually infinite).

5 Conclusion and Future Work

To summarize, we proposed in this paper an extension of the DET-POMDP framework to stochastic observability, called QDET-POMDP, that bridges a part of the gap between DET-POMDPs and general POMDPs. A study of their convergence properties leads to a significant improvement in terms of computational complexity. A sketch of an algorithm is also proposed, opening the avenue of multiple applications. As future work, many avenues can be explored. First, efficient and specific algorithms could be developed that exploit the determinism of transitions and error bounds can be found using the presented bounds on the horizon. Second, an extension to the multiagent case can also lead to major improvements in terms of complexity. Finally, adding some white noise on the transition may help to find more general but still tractable models.

References

1. Nau, D., Ghallab, M., Traverso, P.: Automated Planning: Theory & Practice. Morgan Kaufmann Publishers Inc., San Francisco (2004)
2. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and Acting in Partially Observable Stochastic Domains. *Artif. Intell.* 101(1-2), 99–134 (1998)

3. Palacios, H., Geffner, H.: From Conformant into Classical Planning: Efficient Translations that May Be Complete Too. In: Proc. of the 17th Int. Conf. on Automated Planning and Scheduling, pp. 264–271 (2007)
4. Amir, E., Chang, A.: Learning Partially Observable Deterministic Action Models. *J. Artif. Intell. Res.* 33, 349–402 (2008)
5. Littman, M.: Algorithms for Sequential Decision Making. PhD thesis, Department of Computer Science, Brown University (1996)
6. Bonnet, B.: Deterministic POMDPs Revisited. In: Proc. of Uncertainty in Artificial Intelligence (2009)
7. Pattipati, K., Alexandridis, M.: Application of Heuristic Search and Info. Theo. to Sequential fault Diagnosis. *IEEE Trans. on Syst., Man and Cyb.* 20(4), 872–887 (1990)
8. Ji, S., Parr, R., Carin, L.: Nonmyopic multiaspect sensing with partially observable markov decision processes. *IEEE Transactions on Signal Processing* 55(6), 2720–2730 (2007)
9. Goldman, R.P., Boddy, M.S.: Expressive planning and explicit knowledge. In: Proc. of the 3rd Inter. Conf. on Artif. Intel. Planning Systems, pp. 110–117 (1996)
10. Sondik, E.J.: The optimal control of Partially Observable Markov Processes. PhD thesis, Stanford University (1971)
11. Papadimitriou, C., Tsisiklis, J.: The Complexity of Markov Decision Processes. *Math. Oper. Res.* 12(3), 441–450 (1987)
12. Stockmeyer, L.J.: The Polynomial-Time Hierarchy. *Theor. Comput. Sci.* 3(1), 1–22 (1976)

Appendix A

Proof (Proof of Theorem 7)

$$\begin{aligned}
 & \frac{\theta^n \frac{(1-\theta)^p}{\nu^p}}{\theta^n \frac{(1-\theta)^p}{\nu^p} + \theta^p \frac{(1-\theta)^n}{\nu^n} + (\nu-1) \frac{(1-\theta)^k}{\nu^k}} \geq 1 - \varepsilon \\
 \Leftrightarrow & \frac{\nu^n \theta^n (1-\theta)^p}{\nu^n \theta^n (1-\theta)^p + \nu^p \theta^p (1-\theta)^n + (\nu-1)(1-\theta)^k} \geq 1 - \varepsilon \\
 \Leftrightarrow & \frac{\nu^p \theta^p (1-\theta)^n}{\nu^n \theta^n (1-\theta)^p} + \frac{(\nu-1)(1-\theta)^k}{\nu^n \theta^n (1-\theta)^p} \leq \frac{1}{1-\varepsilon} - 1 \\
 \Leftrightarrow & \nu^{k-2n} \theta^{k-2n} (1-\theta)^{2n-k} + (\nu-1) \frac{\theta^{-n} \nu^{-n}}{(1-\theta)^{-n}} \leq \frac{\varepsilon}{1-\varepsilon} \\
 \Leftrightarrow & \frac{\nu^{k-2n} \theta^{k-2n}}{(1-\theta)^{k-2n}} \left[1 + (\nu-1) \frac{\nu^{-p} \theta^{-p}}{(1-\theta)^{-p}} \right] \leq \frac{\varepsilon}{1-\varepsilon} \\
 \Leftrightarrow & (k-2n) \ln \frac{\nu \theta}{(1-\theta)} + \ln \left[1 + (\nu-1) \frac{\nu^{-p} \theta^{-p}}{(1-\theta)^{-p}} \right] \leq \ln \frac{\varepsilon}{1-\varepsilon} \\
 \Leftrightarrow & (k-2n) \ln \frac{\nu \theta}{(1-\theta)} \leq \ln \frac{\varepsilon}{1-\varepsilon} - \ln \left[1 + (\nu-1) \frac{(1-\theta)^p}{\nu^p \theta^p} \right] \\
 \Leftrightarrow & (2n-k) \ln \frac{\nu \theta}{(1-\theta)} \geq \ln \frac{1-\varepsilon}{\varepsilon} + \ln \left[1 + (\nu-1) \frac{(1-\theta)^p}{\nu^p \theta^p} \right] \\
 \Leftrightarrow & (2n-k) \geq \frac{\ln \frac{1-\varepsilon}{\varepsilon}}{\ln \frac{\nu \theta}{(1-\theta)}} + \frac{1}{\ln \frac{\nu \theta}{(1-\theta)}} \ln \left[1 + (\nu-1) \frac{(1-\theta)^p}{\nu^p \theta^p} \right] \\
 \Leftrightarrow & n \geq \frac{\ln \frac{1-\varepsilon}{\varepsilon}}{2 \ln \frac{\nu \theta}{(1-\theta)}} + \frac{1}{2 \ln \frac{\nu \theta}{(1-\theta)}} \ln \left[1 + (\nu-1) \frac{(1-\theta)^p}{\nu^p \theta^p} \right] + \frac{k}{2} \tag{7}
 \end{aligned}$$

but since $2 \leq \nu \leq |\mathcal{S}| - 1$, $n > \frac{k}{2}$ and $\frac{1-\theta}{\theta} < 1$,

$$\begin{aligned}
 \ln \left[1 + \frac{|\mathcal{S}| - 2}{\nu^{k-n}} \frac{(1-\theta)^{k-n}}{\theta^{k-n}} \right] & \leq \ln \left[1 + \frac{|\mathcal{S}| - 2}{\nu^{\frac{k}{2}}} \left(\frac{1-\theta}{\theta} \right)^{\frac{k}{2}} \right] \\
 & \leq \ln \left[1 + \nu^{1-\frac{k}{2}} \right]
 \end{aligned}$$

From which, ensuring Eqn. (3) also ensures Eqn. (7),

Hierarchical Text Classification Incremental Learning

Shengli Song, Xiaofei Qiao, and Ping Chen

Software Engineering Institute, Xidian University, 710071 Xi'an, China
shlsong@xidian.edu.cn

Abstract. To classify large-scale text corpora, an incremental learning method for hierarchical text classification is proposed. Based on the deep analysis of virtual classification tree based hierarchical text classification, combining the two application models of single document adjustment after classification and new sample set learning, a dynamic online learning algorithm and a sample set incremental learning algorithm are put forward. By amending classifiers and updating the feature space, the algorithms improve the current classification models. Hierarchical text classification experiments on Newsgroup datasets show that the algorithms can enhance the classification accuracy effectively and reduce the storage space and the learning time cost of the history sample datasets.

Keywords: machine learning, incremental learning, hierarchical text classification, text mining.

1 Introduction

Text classification is the automatic assignment of a category label to a text document. Classification, a widely used means to organize a large number of text information, is able to position the information in an accurate and effective way and makes information browsing and searching handily. A number of classification methods have been applied to text classification, including nearest neighbor classification, neural networks, derived probability classification, Boosting and SVM (Support Vector Machine, SVM), etc. [1]. Using the hierarchical relationships among categories and combining with divide-and-conquer thinking, the hierarchical classification decomposes large-scale classification problems. The hierarchical classification, non-linear, data skew and tagging bottleneck are the key questions in current text classification field [2]. Because of the advantages of maintaining accuracy, reducing history data storage space and significantly decreasing the data learning time, the incremental learning method has become the key technology in the intelligent information discovery when dealing with the rapid updating digital text information.

Incremental learning methods are generally applicable for the case involving very large amounts of data or the ever-changing data, such as the log data and the intelligence data. Decision tree and neural network are the mostly used algorithms for the realization of the existing incremental learning methods [3], [4], [5]. These realizations have some problems to certain degrees in practice: the lack of anticipant risk control to the entire sample set causes over fitting; the lack of selective oblivion and

elimination mechanism for the training data affects the classification accuracy to a large extent. The related research has been addressing by many researchers [6], [7], [8], and [9]. However, most of the methods above do not involve the hierarchical structure and most of them apply to the small support vector set or need to select many parameters values of which are uncertain [10]. Thus follow-up classification results are affected. It is better to adjust the hierarchical structure among categories than those methods mentioned above; the hierarchical classification will receive wilder applications. However, due to the difference between the hierarchical classification approach and traditional single-level one, the incremental learning methods requires re-design in accordance with the characteristics of the categories. Study of the incremental learning method for the hierarchical classification is relatively little by now.

On the base of hierarchical text classification approach and incremental learning algorithms available, dynamic online learning and batch incremental learning algorithms for hierarchical text classification are proposed for two application schemas of single document adjustment after classification and new sample set learning. The experiment indicates that the algorithm is effective in enhancing the classification accuracy and reduces the learning time cost on the history sample datasets.

2 Virtual Classification Tree

In the learning process of text classification the potential associations existing between documents and categories are extracted from training data collection by the probability statistics or machine learning algorithm, and the classification model is generated for the follow-up classification process. The classification tree [12], [15] and the directed acyclic graph (including Yahoo! and Open Directory Project, etc.) are two basic forms of categories applied in proposed hierarchical text classification approaches [11], [12], [13], [14], [15], [16]. In the former form the categories of documents is structured as a tree and categories is top-down determined level by level in the classification process; In the latter form nodes in the directed acyclic graph is travel searched based on graphic theory and method of directed acyclic graph for the determination of document classifications. A virtual class tree, proposed by S. Dumais and H. Chen [12], is used in this paper to represent the classification model, the leaf-nodes of which express categories.

Definition 1. VCTree is defined as $(N^{root}, \{N_{ij}^{inner}\}, \{N_k^{class}\}, \{(N_f, N_s)\})$.

- N^{root} is the ancestor of all nodes, namely, a virtual category aggregated by all categories;
- N_{ij}^{inner} is the j^{st} virtual category at the depth of i in the VCTree, represented as the all non-leaf nodes except the root node;
- N_k^{class} is the k^{st} actual category, represented as the leaf nodes in VCTree;
- (N_f, N_s) is the father-son relationship between nodes, represented as the branches in VCTree. $N_f \in \{N^{root}\} \cup \{N_{ij}^{inner}\}$, $N_s \in \{N_{ij}^{inner}\} \cup \{N_k^{class}\}$.

An instance of VCTree is shown in Fig.2. There are three types of nodes:

- Root node: *Newsgroup* represents virtual categories of all the documents;
- Inside nodes: $\{alt, comp, rec, talk, sci\}$ is virtual category set;
- Category nodes: $\{atheism, graphics, windows, autos, motorcycles, politics, religion, crypt, electronics, med, space\}$ is actual category set to which the documents are to be assigned.

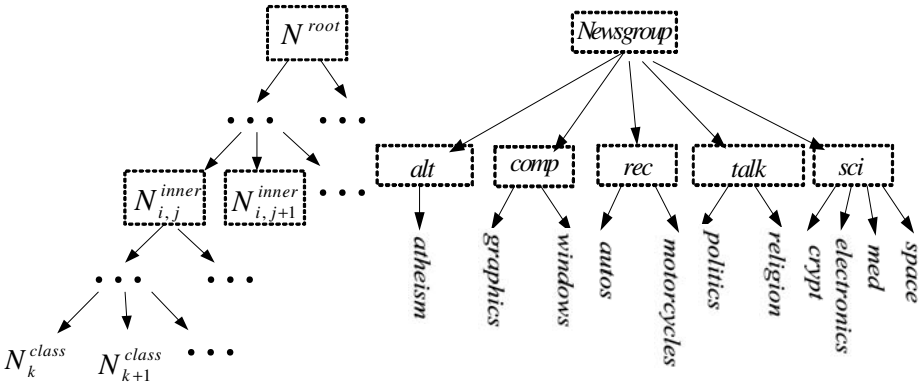


Fig. 1. The structure of virtual category tree Fig. 2. Sample of two level virtual category tree

3 Incremental Learning for Hierarchical Text Classification

In the running process of the text classification systems, the semantic center of the non-categorized document set changes with the passage of time and the creation number of samples compared with the original training document set, but the classification model remains, thus the accuracy of the classification is reduced. One solution is to re-learn the classification model. While it is difficult to construct a large-scale sample set and keep the forward compatibility of the classification model at the same time. Besides large time overhead, the effect is not very satisfactory.

Incremental learning is able to modify the classification model through a continuous addition of new sample set in the using of the text classification system, so as the accuracy of classification is achieved.

Though an analysis of the application environment of the incremental learning in practice, we have found that two situations are suitable for the incremental learning algorithm:

- 1) Incremental learning for one single document: class modification occurs in single document. When document d_c is categorized into class N_m^{class} by a mistake, domain experts will modify the class as N_n^{class} , thus an incremental learning behavior is arose.
- 2) Incremental learning for document set: New learning sample set is brought in. History learning sample set has a corresponding classification model VCTree. New arrived learning sample set needs a new classifier VCTree* built on the base of VCTree, which resulting in the incremental learning behavior.

3.1 Dynamic Online Learning

Because the accuracy of the current text classification system could not be guaranteed completely, the domain experts will usually modify the mistaken documents after the automatic classification by the system. The modified result is hoped to work as a learning sample for the performance improvement of the current system. According to SVM classification theory, most of these modified documents are usually at the position near by the hyper-plane. If those documents could be learning as samples and corresponding feature set as the support vectors of the classification model, a slight moving of the SVM classification model will generated by the modification of those documents.

An effect of one single modified document on the classification model is revealed in

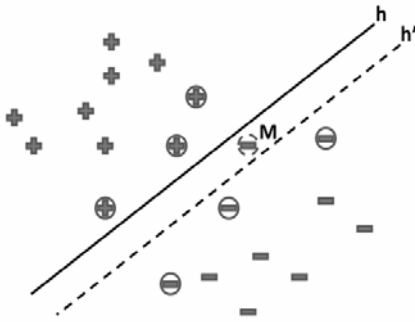


Figure 3. In the linearly separable space, h expresses the hyper-plane of the classification model, document M was categorized into a category below h , the experts then adjusted M to the category above h . If M is learned as a sample, hyper-plane h will move to h' , thus causing changes in the classification model. If M is used as a test samples, it will be categorized into the category above h next time. Consequently classification result will be more accurate.

Fig. 3. Moving of SVM hyper-plane

An adjustment for one document category corresponds an adjusting of the node classification path on VCTree, which means to transfer one document from one leaf node to another at other branches. History sample set is signed as D_{train} , the modified document is signed as $d_c (d_c \notin D_{train})$, N_m^{class} and N_n^{class} are two leaf nodes with which D_{train} is corresponds in VCTree. d_c is adjusted from the category of N_m^{class} to that of N_n^{class} and the binary classifier are re-trained. VCTree* is updated according to algorithm 1.

Definition 2. Class path sequence is expressed as the virtual category list of the direct ancestor nodes of one category in VCTree in order of a top-down sequence. The class path sequence of N_t^{class} is:

$$Path(N_t^{class}) = \langle N_{root}, N_{i_1}, \dots, N_{m_i}, N_t^{class} \rangle$$

$$N_{ij} \in \{N^{inner}\}; m \text{ is the depth of the father node of } N_t^{class} \text{ in VCTree.}$$

Definition 3. Correct (Wrong) sub-path sequence $Tpath (Fpath)$ is the path composed by the nodes in the correct (wrong) class path of the document without existing in the wrong (correct) class path. They can be expressed as:

$$Tpath(m, n) = Path(N_n^{class}) - (Path(N_m^{class}) \cap Path(N_n^{class}))$$

$$Fpath(m, n) = Path(N_m^{class}) - (Path(N_m^{class}) \cap Path(N_n^{class}))$$

$Path(N_n^{class})$ is the correct class path, $Path(N_m^{class})$ is the wrong class path.

Definition 4. Path node function $first(path)$ is defined as the calculation of the first node return of the path.

Algorithm 1. Single Document Adjustment Incremental Learning

Input: VCTree, document d_c , source class node N_m^{class} and target node N_n^{class} .

Output: VCTree*.

Step 1. Depth-first search of VCTree according to N_m^{class} and N_n^{class} respectively in order to determine class path sequences $Path(N_m^{class})$ and $Path(N_n^{class})$;

Step 2. Define variable $tpath = Tpath(n, m)$, $fpath = Fpath(m, n)$;

Step 3. If $tpath \neq \langle \rangle \cap fpath \neq \langle \rangle$, implement following steps in a loop:

- a) If $tpath \neq \langle \rangle$, add d_c as a positive sample into support vector set and re-train the binary classifier with which the first node corresponds, $tpath = Tpath(n, m) - first(Tpath(n, m))$;
- b) If $fpath \neq \langle \rangle$, add d_c as a negative sample into support vector set and re-train the binary classifier with which the first node corresponds, $fpath = Fpath(m, n) - first(Fpath(m, n))$;

Step 4. Output VCTree*, the one built after the update of the node classifier.

The re-train process to N_n^{class} corresponding binary classifier is to change the relative support vector coefficient so as for the original support vectors to meet the KKT constraint condition. The differential equation for KKT is expressed as:

$$\Delta L_i = Q_{ic} \Delta \alpha_i + \sum_{d_j \in S} Q_{ij} \Delta \alpha_j + y_i \Delta b, \forall d_i \in S \cup \{c\} \tag{1}$$

$$y_c \Delta \alpha_c + \sum_{d_j \in +sv} y_j \Delta \alpha_j = 0 \tag{2}$$

S is the history sample set, C is newly added sample vector, α_c is the coefficient with the initial value of 0. +sv is the boundary support vector. When the non-boundary support vector for the history sample vectors $-sv = (v_1, v_2, \dots, v_m)$ and $L_i \equiv 0$, differential equation (1) can be expressed as a matrix:

$$\begin{bmatrix} 0 & y_{v_1} & \dots & y_{v_m} \\ y_{v_1} & Q_{v_1 v_1} & \dots & Q_{v_1 v_m} \\ \vdots & \vdots & \ddots & \vdots \\ y_{v_m} & Q_{v_m v_1} & \dots & Q_{v_m v_m} \end{bmatrix} \begin{bmatrix} \Delta b \\ \Delta \alpha_{v_1} \\ \vdots \\ \Delta \alpha_{v_m} \end{bmatrix} = - \begin{bmatrix} y_c \\ Q_{v_1 c} \\ \vdots \\ Q_{v_m c} \end{bmatrix} \Delta \alpha_c \tag{3}$$

Let Q be the (m + 1)-order symmetric non-positive definite Jacobian matrix of the left side of the equation. For the balance keeping, let:

$$\Delta b = \beta \Delta \alpha_c, \Delta \alpha_j = \beta_j \Delta \alpha_c, d_j \in -sv \tag{4}$$

β, β_j are the coefficient sensitivity. For $d_j \notin -sv$ there is $\beta_j \equiv 0$. Equation (2) can be transferred as:

$$\begin{bmatrix} \beta \\ \beta_{v_1} \\ \vdots \\ \beta_{v_m} \end{bmatrix} = -Q^{-1} \begin{bmatrix} y_c \\ Q_{v_1c} \\ \vdots \\ Q_{v_m c} \end{bmatrix} \tag{5}$$

Bringing equation (4) in to equation (1), following equation is got:

$$\Delta L_i = g_i \Delta \alpha_c \quad \forall d_i \in S \cup \{c\} \tag{6}$$

g_i is the boundary sensitivity, which can be expressed as:

$$g_i = Q_{ic} + \sum_{d_j \in +sv} Q_{ij} \beta_j + y_i \beta \quad \forall d_i \notin +sv \tag{7}$$

For $i \in -sv, L_i \equiv 0$, so $g_i \equiv 0$. Adding non-boundary support vector $-sv$ into c, Q^{-1} is updated as:

$$Q = \begin{bmatrix} & & 0 & 0 \\ & Q^{-1} & \vdots & 0 \\ & & 0 & \vdots \\ 0 & \dots & 0 & 0 \end{bmatrix} + \frac{1}{g_c} \begin{bmatrix} \beta \\ \beta_v \\ \vdots \\ \beta_{v_m} \\ 1 \end{bmatrix} \cdot [\beta \quad \beta_{v_1} \quad \dots \quad \beta_{v_m} \quad 1] \tag{8}$$

g_c, β, β_i can be obtained from equation (4) and (7).

3.2 Batch Incremental Learning

With the increasing number of documents, changes will happen to the semantic center of each category, which result a reduction in the accuracy of the classification system. Through a volume increase in learning samples, the category model could be adjusted to the semantic center of the samples; therefore the accuracy of the classification system can be maintained or even increased.

During the incremental learning process, new document vector will occur in the documents feature vector set of the sample set, which is able to work as a support vector of the category model to impact the classification result. Therefore, the effect of the sample set incremental learning for single document cannot be revealed merely by means of adjusting the values of the support vectors. A newly re-built support vector set of the classification model and a change to the original vector space are needed in order to enhance the accuracy of the classification. Feature selection and hierarchical incremental learning are involved in sample set incremental learning.

For incremental learning, a judgment that whether the KKT condition is satisfied is carried out to newly added sample set by the benefit of the original classification model. The sample meeting the KKT condition will not change the support vector set of the classifier, while the one contrary to the KKT conditions will change the set. $f(x)$ is the SVM classification decision function, newly added sample $d_j = (x_c, y_c)$, x_c, y_c are the hyper-plane coordinates of d_c . Classification interval is $[-1, 1]$. Classification hyper-plane is $f(x) = 0$. Samples that contrary to the KKT condition are divided into three categories:

- 1) Samples are located in the classification interval and in the same side of the classification interval with their categories. These samples are correctly categorized by the original classifier, and satisfied $0 \leq y_c, f(x_c) < 1$;
- 2) Samples are located in the classification interval and in the other side of the classification interval with their categories. These samples are not correctly categorized by the original classifier, and satisfied $-1 \leq y_c, f(x_c) < 0$;
- 3) Samples are located out of the classification interval and in the other side of the classification interval with their categories. These samples are not correctly categorized by the original classifier, and satisfied $y_c f(x_c) < -1$.

Considering the fact that samples which satisfied the KKT condition could be correctly categorized, a neglect of the impact of these samples on the classification model is adopted in this paper. Samples that violate the KKT condition and could result in classification errors are used for the incremental learning. In this way, the number of samples is effectively reduced and the incremental learning is accelerated.

One important factor that impacts the accuracy of the classification system during its running process is the inaccuracy of the support vector set of the classification model. Based on the support vector set of the classification model, incremental feature selection is the increment of high weight support vectors and the elimination of part of the low weight support vectors in order to form a new support vector set.

Feature words selection is an important factor which affects the accuracy of text classification. Especially in the incremental learning environment, with the continuously addition of documents, changes will happen in feature space set and weights, while sample data of the original feature space cannot be added in for calculation. So before the incremental learning, an incremental feature selection is necessary to adjust the effect of the feature space on the incremental learning.

Definition 5. Term word FW is defined as a five- dimension array (ID, TX, TF, DF, WT) :

- $TX \in Dict$; $Dict$ is the word item set in the dictionary;
- $ID = Dict(TX) \in \mathbb{N}$; $Dict(TX)$ is the identifier of TX in the dictionary;
- $TF = (tf_1, tf_2, \dots, tf_K), tf_i = \sum_{d_j \in N_i^{class}} |TX_{d_j}|$; TF is the frequency of word TX in every category training document set, namely the frequency in which the word TX occurs in document set; K is the number of the actual categories;
- $DF = (df_1, df_2, \dots, df_K), df_i = |\{d_j \mid d_j \in N_i^{class} \cap TX \in d_j\}|$; DF is the frequency of documents in each category in which the word TX is exist, namely, the number of documents the feature words of which have word TX ;
- $WT = Dictf(TX)$; $Dictf(TX)$ is the frequency factor of the word TX in the document.

Definition 6. Feature space F_{space} is defined as the set of the key feature words; the key feature words are selected from the training document set by the weight calculation through information gain or mutual information approach.

Definition 7. Feature word set of document d_j

$S^{d_j} = \{(fw_i, v_i) \mid fw_i, tx \in d_j, v_i = fw_i \cdot tf \cdot \log(|D^{train}| / fw_i \cdot df)\}; fw_i : FW\}$, v_i is the weight of the feature word fw_i in document d_j through the calculation of $tf \cdot idf$.

Definition 8. Feature word set of sample set D $S^D = \{S^{d_j} \mid d_j \in D\}$.

Algorithm 2. Document Feature Incremental Selection

Input: Feature space F_{space} and fresh sample set D_{new}

Output: Updated feature space F_{space}^*

Step 1. Execute hierarchical HMM-based Chinese word segmentation and dimensionality reduction to all the samples in fresh sample set D_{new} , obtain the new feature word set $S^{D_{new}}$ of D_{new} ;

Step 2. For each $d_j \in D_{new}$, define document count sign $\Delta_{df} = false$:

For each $(fw_i, c_i) \in S^{d_j}$:

If there is $fw_i \in F_{space}$ and $fw_i.id = fw_i.id$ is satisfied, then $fw_i.tf += c_i$;
 $fw_i.df += 1$ if $\Delta_{df} = false$;

Else, define new feature word $fw_{new} = \langle fw_i.id, fw_i.tx, 1, 1, fw_i.wt \rangle$;

$F_{space} = F_{space} \cup \{fw_{new}\}$.

Step 3. For each feature word in F_{space} , calculate the weight by the feature selection algorithm; re-select features to get new feature space F_{space}^* .

Fresh sample set has been addressed by incremental feature selection to form a new feature space. Weights of the feature words in original feature space need a second calculation to adjust the new feature space. Fresh sample set incremental learning is described in Algorithm 3.

Definition 9. Feature space mapping function is defined as:

$$\Phi(fw_i \in F_{space}, fw_i^* \in F_{space}^*) = \{fw_i.id = fw_i^*.id : fw_i.tf + = fw_i^*.tf, fw_i.df + = fw_i^*.df\} \quad (9)$$

In the above function, key feature word set in original feature space is united with the fresh one. Feature words are selected or eliminated according to the value of weight. The weight of remaining feature words is the summation of the history weight and new one.

Algorithm 3. Fresh Sample-Set Incremental Learning

Input: VCTree, fresh sample set D_{new}

Output: VCTree*, the updated VCTree

Step 1. Update the feature word space according to D_{new} by incremental feature selection algorithm;

Step 2. Introduce feature space mapping function $\Phi: F_{space} \rightarrow F_{space}^*$, re-calculate weights of feature words in feature space F_{space} of VCTree using $tf \cdot idf$ and map them to new feature space F_{space}^* ;

- Step 3. Top-down update each classifier of each node in VCTree according to the result of re-calculation to feature word id and wt ;
- Step 4. Generate $VCTree^*$ based on F_{space}^* .
-

By the second feature selection, fresh sample set incremental learning algorithm has the ability to combine history samples with fresh ones and re-build VCTree. Performance of the hierarchical text classification is guaranteed. Furthermore, sample set incremental learning algorithm saves the storage space and speed up training to classifiers by the benefit of non-keeping of history sample set as while as reduction of time overhead for repeat learning.

4 Experiment and Result Analysis

4.1 Experiments Setup

Newsgroup20 is used as the data set in the following experiment. The category structure is revealed in figure 2. Preprocess is completed including text extraction, which is done by semantic structure-based web page text extraction tool developed by the text processing research group of Software Engineering Institute of Xidian University, and Chinese word segmentation, which is done by Chinese word segmentation system ICTCLAS developed by Computing Technology Institute of Chinese Scientific Research Institute. SVM has been proved to be an effective way for text classification [18], [19]. Dumais and Chen have proved the good performance of SVM in classification tree structures [12]. SVMlight classifier [20], developed by Joachims is integrated in the system.

Throughout the experiments, information gain is used for feature selection. We use TF-IDF to compute the feature's weight. The size of feature space is 1000, which means we choose 1000 features globally. Linear kernel functions are chosen for SVMs, in which the penalty parameter $C=1$. The precision rate, recall rate and F_1 value are used to evaluate the classification result.

4.2 Experiments Results

Experiment 1. Incremental learning after single document adjustment. According to the structure revealed in figure 2, 300 documents were selected as the training sample set for each category in Newsgroup, while another 700 documents as the test sample set for each category. After the classification process, documents {53150, 53152, 53153, 53155, 53156}, which were wrongly classified into talk.religion, are dragged into the correct category alt.atheism through the GUI operation. This behavior triggered the single document adjustment incremental learning for five times, followed by re-classification for the same test sample set. A comparison for the two classification result is revealed in table 1.

Table 1. Results of dynamic online learning of single document adjustment

Category	Documents		Before adjustment(%)			After adjustment(%)		
	Train	Test	Recall	Precision	F ₁	Recall	Precision	F ₁
alt.atheism	300	700	68.6	94.3	79.4	68.9	94.1	79.6
comp.graphics	300	700	95.4	97.5	96.5	95.4	97.5	96.5
comp.windows	300	700	97.6	95.3	96.4	97.6	95.3	96.4
rec.autos	300	700	97.9	98.8	98.3	97.9	98.8	98.3
rec.motorcycles	300	700	99.4	99.0	99.2	99.4	99.0	99.2
sci.crypt	300	700	98.6	98.6	98.6	98.6	98.6	98.6
sci.electronics	300	700	97.6	97.9	97.7	97.6	97.9	97.7
sci.med	300	700	98.4	98.4	98.4	98.4	98.7	98.6
sci.space	300	700	98.1	98.7	98.4	98.1	98.7	98.4
talk.politics	300	700	83.3	95.7	89.1	83.3	95.7	89.1
talk.religion	300	700	94.6	66.9	78.3	94.6	67.0	78.4
ALL	3300	7700	93.6	94.6	93.6	93.6	94.7	93.7

Table 2. Results of fresh sample set incremental learning

Category	First (200+)(%)			Second (200+)(%)		
	Recall	Precision	F ₁	Recall	Precision	F ₁
alt.atheism	74.7	91.8	82.4	78.6	90.6	84.2
comp.graphics	96.7	97.1	96.9	97.7	97.4	97.6
comp.windows	97.1	96.9	97.0	98.0	97.9	97.9
rec.autos	99.3	98.0	98.7	99.7	99.1	99.4
rec.motorcycle	99.3	99.7	99.5	100	100	100
sci.crypt	99.3	98.4	98.9	99.6	98.9	99.2
sci.electronics	98.0	99.3	98.6	98.1	99.9	99.0
sci.med	98.6	98.4	98.5	99.4	98.0	98.7
sci.space	98.1	98.6	98.4	99.4	99.0	99.2
talk.politics	86.9	94.6	90.5	88.6	95.2	91.8
talk.religion	89.4	70.7	79.0	87.4	73.9	80.1
ALL	94.3	94.9	94.4	95.1	95.5	95.2

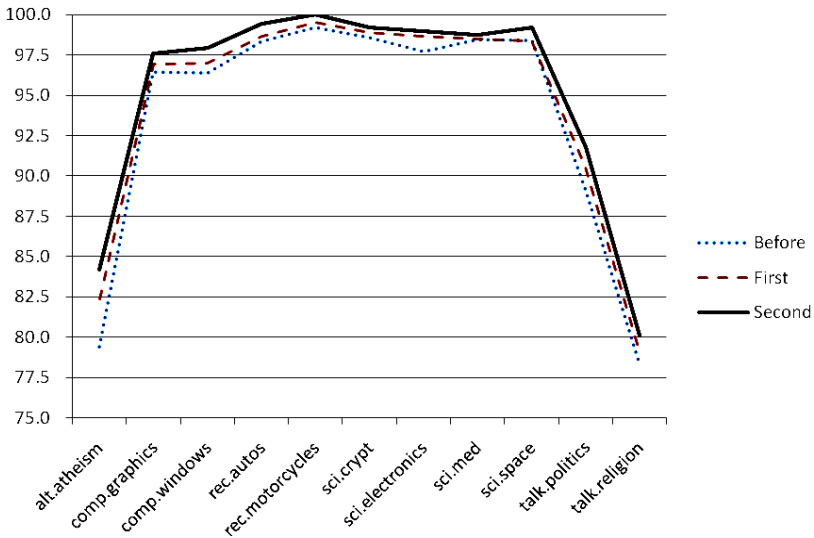


Fig. 4. Comparison between two steps of sample dataset incremental learning

Table 1 shows that the classification model has been improved by the effort of the dynamic single document adjustment, and the recall rate, the precise rate and the F_1 value of the classification have been elevated slightly.

Experiment 2. Sample set incremental learning. Based on the experiment 1, the incremental learning method is validated by the two-step incremental learning. The first step of the this validation is to add 200 documents to the original training set which contained 300 documents of each category, and then the re-classification is carried out. The second step is similar to the first step, except for the augment of the newly 200 training documents to 500 ones. Results of these two classifications after the incremental learning is shown in table 2. Data in this table indicates that the F_1 value was increased from 93.6% to 94.4% after the first step learning, and the F_1 value was increased from 94.4% to 95.2% after the second step. Performance of the before and after incremental learning tests is given in figure 4. To some degree, the precision rate and recall rate of other categories are enhanced.

4.3 Results Analysis

Results from Experiment 1 and Experiment 2 indicate that single document or fresh sample set incremental learning will impact the performance of the hierarchical classification in a positive way. Single document incremental learning will make the hyper-plane of SVM binary classifier move slightly. In the following classification, the document that has the similar semantic center with the learned single document will be correctly categorized, thus improvement on the classifier is completed. The feature re-selection in fresh sample set is to make the history samples a representation by the feature item index in classifiers and a compare with the feature representation of the fresh sample set. By the re-selection and elimination of feature items, the classification feature space will be more accurate. The classification accuracy will be guaranteed without a re-learning of history samples.

Through the experiment for the first incremental learning, the time we need is 34.4s ($F_1=94.4\%$). If we train all the dataset including history and fresh samples, the time consumed is 44.6s ($F_1=94.6\%$). For the second incremental learning experiment, the time cost can be decreased by 32.4%, that are 39.4s ($F_1=95.2\%$) and 58.3s ($F_1=95.4\%$) respectively. So, the incremental learning can reduce the time cost evidently.

5 Conclusion

The existing text classification approach is analyzed and the incremental learning method is designed in this paper. Elevating the classification performance, incremental learning algorithm reduces the storage space for history samples and save the time for re-learning of historical samples. Experiment shows that the hierarchical classification approach and the incremental learning method could achieve the desired objective and meet the requirements in applications. In future work, the tolerant hierarchical classification approach and the balance algorithm for VCTree would be main research directions.

References

1. Cai, L., Hofmann, T.: Hierarchical Document Classification with Support Vector Machines. In: Proceedings of the thirteenth ACM international conference on information and knowledge management, pp. 78–87 (2004)
2. Jinshu, S., Bofeng, Z., Xin, X.: Advances in Machine Learning Based Text Classification. *Journal of Software* 17(9), 1848–1859 (2006)
3. Ratsaby, J.: Incremental Learning with Sample Queries. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(8), 883–888 (1998)
4. Yamauchi, K., Ishii, N.: Incremental Learning Methods with Retrieving of Interfered Patterns. *IEEE Transactions on Neural Networks* 10(11), 1351–1365 (1999)
5. Pin, T., Bo, Z., Zhen, Y.: An Incremental BiCovering Learning Algorithm for Constructive Neural Network. *Journal of Software* 14(2), 194–201 (2003)
6. Syed, N., Liu, H., Sung, K.: Handling Concept Drifts in Incremental Learning with Support Vector Machines. In: Proceedings of the International Joint Conference on Artificial Intelligence, Stockholm, Sweden (1999)
7. Ruping, S.: Incremental Learning with Support Vector Machines. In: The First IEEE International Conference on Data Mining, pp. 641–642 (2001)
8. Cauwenberghs, G., Poggio, T.: Incremental and Decremental Support Vector Machine Learning. *Machine Learning* 44(13), 409–415 (2001)
9. Fei, W., Dayou, L., Songxin, W.: Research on Incremental Learning of Bayesian Network Structure Based on Genetic Algorithms. *Journal of Computer Research and Development* 42(9), 1461–1466 (2005)
10. Rong, X., Jicheng, W., Zhengxing, S., Fuyan, Z.: An Incremental SVM Learning Algorithm α -ISVM. *Journal of Software* 12(12), 1818–1824 (2001)
11. D’Alessio, S., Murray, K., Schiaffino, R., Kershenbaum, A.: The Effect of Using Hierarchical Classifiers in Text Classification. In: Proc. of the 6th Int. Conf. on Recherche d’Information Assistée par Ordinateur, Paris, FR, pp. 302–313 (2000)
12. Dumais, S., Chen, H.: Hierarchical classification of Web content. In: Proc. of the 23rd ACM Int. Conf. on Research and Development in Information Retrieval, Athens, GR, pp. 256–263 (2000)
13. Mladenic, D.: Turning Yahoo to Automatic Web-page Classifier. In: Proc. of the European Conf. on Artificial Intelligence, pp. 473–474 (1998)
14. Sasaki, M., Kita, K.: Rule-based Text Classification using Hierarchical Categories. In: Proc. of the IEEE Int. Conf. on Systems, Man, and Cybernetics, La Jolla, US, pp. 2827–2830 (1998)
15. Wang, K., Zhou, S., He, Y.: Hierarchical Classification of Real Life Documents. In: Proc. of the 1st SIAM Int. Conf. on Data Mining, Chicago (2001)
16. Weigend, S., Wiener, D., Pedersen, J.: Exploiting Hierarchy in Text Classification. *Information Retrieval* 1(3), 193–216 (1999)
17. ODP-Open Directory Project, <http://dmoz.org/>
18. Dumais, S., Platt, J., Heckerman, D., Sahami, M.: Inductive Learning Algorithms and Representations for Text Classification. In: Proc. of the 7th Int. Conf. on Information and Knowledge Management, pp. 148–155 (1998)
19. Joachims, T.: Text Classification with Support Vector Machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
20. Joachims, T.: An implementation of Support Vector Machines (SVMs) in C, <http://ais.gmd.de/~thorsten/svmlight/>

Robust Stability of Stochastic Neural Networks with Interval Discrete and Distributed Delays

Song Zhu, Yi Shen*, and Guici Chen

Department of Control Science and engineering,
Huazhong University of Science and Technology, 430074 Wuhan, China
Tel.: +86 27 87543630; Fax:+86 27 87543631
yishen64@163.com, zhusonghust@smail.hust.edu.cn

Abstract. This paper is concerned with the global asymptotic stability analysis problem for a class of stochastic neural networks with interval discrete and distributed delays. The parameter uncertainties are assumed to be norm bounded. Based on Lyapunov-Krasovskii stability theory and the stochastic analysis tools, sufficient stability conditions are established by using an efficient linear matrix inequality(LMI) approach. It is also shown that the result in this paper cover some recently published works. A numerical example is provided to demonstrate the usefulness of the proposed criteria.

Keywords: Stochastic neural networks, Lyapunov functional, Linear matrix inequality, asymptotic stability.

1 Introduction

In the past 20 years, neural networks have found fruitful applications in numerous areas such as combinatorial optimization, signal processing, pattern recognition and many other fields [12,13,20]. However, all successful applications are greatly dependent on the dynamic behaviors of neural networks. As is well-known now, stability is one of the main properties of neural networks, which is a crucial feature in the design of neural networks. On the other hand, time-delays have been known to exist naturally in neural processing and signal transmission, and are frequently the sources of instability. Various types of time-delays have been investigated, including constant or time-varying delays, discrete and distributed delays, see for example [1,4,22], and the references therein. The corresponding stability criteria can be classified as delayed-independent or delay-dependent conditions, and the former is more conservative than the latter especially for small size delays.

It is worth noting that in real nervous systems, the synaptic transmission is a noisy process brought on by random fluctuations from the release of neurotransmitters and other probabilistic causes. It has been revealed in [3] that a neural network could be stabilized or destabilized by certain stochastic inputs. Consequently, the stochastic stability analysis problem for various neural networks

* Corresponding author.

has stirred increasing research interests, and relevant results have begun to be published [5,7,8,10,11,18]. On the other hand, in hardware implementation of neural networks, the network parameters of the neural system may be subjected to some changes due to the tolerances of electronic components employed in the design. Therefore, it is important to investigate the robust stability of neural networks with parameter uncertainties.

It is known that both discrete and distributed delays should be taken into account when modeling a realistic neural network [14-16,19,21]. Most of the existing results related to time-varying delay systems are based on the assumption $0 < d(t) \leq h_2$, where $d(t)$ is delay and h_2 is a constant. However, in many practical systems, the typical delay may exist in an interval $(0 < h_1 \leq d(t) \leq h_2)$. Typical examples of systems with interval time-delaying delay are cellular neural networks [8-10].

In this paper, we investigate the global asymptotic stability analysis problem for a class of uncertain stochastic neural networks with interval discrete and distributed time-delays. The parameter uncertainties are norm-bounded. Different from the commonly used matrix norm theories, a unified linear matrix inequality(LMI) approach is developed to establish sufficient conditions for neural networks to be globally, asymptotically stable. Note that LMIs can be easily solved by using the Matlab LMI toolbox, and no tuning of parameters is required [2,6]. An example is provided to show the usefulness of the proposed global stability condition.

Notations: The notations are quite standard. Throughout this paper, $|\cdot|$ is the Euclidean norm in R^n . $\lambda_{max}(A)$ (respectively, $\lambda_{min}(A)$) means the largest (respectively, smallest) eigenvalue of A . Moreover, let $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \geq 0}, P)$ be a complete probability space with a filtration $\{\mathcal{F}_t\}_{t \geq 0}$ satisfying the usual conditions (i.e. the filtration contains all p-null sets and is right continuous). The shorthand $\text{diag}\{M_1, M_2, \dots, M_n\}$ denotes a block diagonal matrix with diagonal blocks being the matrices M_1, M_2, \dots, M_n . The notation \star always denotes the symmetric block in one symmetric matrix.

2 Problem Formulation

Consider the following stochastic neural networks with interval discrete and distributed delays can be described by:

$$\begin{aligned} dx(t) = & \left[- (A + \Delta A)x(t) + (W_0 + \Delta W_0)f(x(t)) + (W_1 + \Delta W_1)f(x(t - d(t))) \right. \\ & \left. + (W_2 + \Delta W_2) \int_{t-\tau}^t f(x(s))ds \right] dt + \sigma(x(t), x(t - d(t)), t)dB(t) \end{aligned} \quad (1)$$

where $x(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$ is the neural state vector, the diagonal matrix $A = \text{diag}[a_1, a_2, \dots, a_n]$ has positive entries $a_i > 0$. W_0, W_1 and W_2 are, respectively, the connection weight matrix, the discretely delayed connection weight matrix, and the distributively delayed connection weight matrix, and

the matrices $\Delta A, \Delta W_0, \Delta W_1, \Delta W_2$ represent the time-varying parameter uncertainties. $f(x(t)) = [f_1(x_1(t)), f_2(x_2(t)), \dots, f_n(x_n(t))]^T$ is the neuron activation functions vector with $f(0) = 0$, and $d(t)$ denotes the discrete time-delay range in interval h_1 to h_2 . $\tau > 0$ is the known distributed time-delay, which is less than h_2 . $\sigma(x(t), x(t - d(t)), t)$ is a matrix valued function, $B(t) = [B_1(t), B_2(t), \dots, B_m(t)]^T \in R^m$ is a Brownian motion defined on a complete probability space (Ω, \mathcal{F}, P) with a natural filtration $\{\mathcal{F}_t\}_{t \geq 0}$. In following, we will use $y(t)$ denotes $x(t - d(t))$. In order to obtain our main results, the assumptions are always made throughout this paper.

Assumption 1. The activation function $f(\cdot)$ is bounded, and satisfy the following Lipschitz condition:

$$|f(x)| \leq |Gx| \quad \forall x \in R^n \tag{2}$$

where $G \in R^{n \times n}$ is a known constant matrix.

Assumption 2. The time delay $d(t)$ is a time-varying differentiable function that satisfies

$$0 < h_1 \leq d(t) \leq h_2, \quad \dot{d}(t) \leq \mu < 1, \tag{3}$$

where h_1, h_2, μ are constants.

Assume that $\sigma(x(t), y(t), t)$ is locally Lipschitz continuous and satisfies the linear growth condition, and the matrices $\Delta A, \Delta W_0, \Delta W_1, \Delta W_2$ are of the following structure:

$$[\Delta A, \Delta W_0, \Delta W_1, \Delta W_2] = MF[N_1, N_2, N_3, N_4] \tag{4}$$

where $A, W_i (i = 0, 1, 2), M, N_j (j = 1, 2, 3, 4)$ are known real constant matrices with appropriate dimensions, and the uncertain matrix F , which may be time-varying, is unknown and satisfies

$$F^T F \leq I \tag{5}$$

Let $x(t, \xi)$ denote the state trajectory of the neural network (1) from the initial data $x(\theta) = \xi(\theta)$ on $-h_2 \leq \theta \leq 0$ in $L^2_{\mathcal{F}_0}([-h_2, 0]; R^n)$. It can be easily seen that the system (1) admits a trivial solution $x(t; 0) \equiv 0$ corresponding to the initial data $\xi = 0$.

Definition 1. For the neural network (1) and every $\xi \in L^2_{\mathcal{F}_0}([-h_2, 0]; R^n)$, the trivial solution is globally asymptotically stable in the mean square if for all admissible uncertainties

$$\lim_{t \rightarrow \infty} E|x(t, \xi)|^2 = 0 \tag{6}$$

3 Main Results and Proofs

The following lemma will be essential in establishing the desired LMI-based stability criteria.

Lemma 1. (i) Let $x \in R^n, y \in R^n$ and $\varepsilon > 0$. Then we have $2x^T y \leq \varepsilon x^T x + \varepsilon^{-1} y^T y$.
 (ii) For any constant matrix $M \in R^{n \times n}, M = M^T > 0$, a scalar $\rho > 0$, vector function $\omega : [a, b] \rightarrow R^n$ such that the integrations are well defined, the following inequality holds:

$$\left(\int_a^b \omega(s) ds \right)^T M \int_a^b \omega(s) ds \leq (b - a) \int_a^b \omega^T(s) M \omega(s) ds.$$

(iii) (Schur complement) Given constant matrices $\Omega_1, \Omega_2, \Omega_3$ where $\Omega_1 = \Omega_1^T$ and $0 < \Omega_2 = \Omega_2^T$, then $\Omega_1 + \Omega_3^T \Omega_2^{-1} \Omega_3 < 0$ if and only if

$$\begin{pmatrix} \Omega_1 & \Omega_3^T \\ \Omega_3 & -\Omega_2 \end{pmatrix} < 0, \quad \text{or} \quad \begin{pmatrix} -\Omega_2 & \Omega_3 \\ \Omega_3^T & \Omega_1 \end{pmatrix} < 0$$

Theorem 1. Assume that there exist matrices $P > 0, D_0 > 0$ and $D_1 > 0$ such that

$$\text{trace}[\sigma^T(x(t), y(t), t) P \sigma(x(t), y(t), t)] \leq x^T(t) D_0 x(t) + x^T y(t) D_1 y(t) \tag{7}$$

the uncertain stochastic neural network(1) is robustly, globally, asymptotically stable in the mean square, if there exist positive definite matrices Q_1, Q_2, Q_3, Z_1, Z_2 , and scalar $\varepsilon_i (i = 1, \dots, 7) > 0$ such that the LMI holds

$$\Pi = \begin{pmatrix} \Pi_1 & \Pi_2^T \\ \Pi_2 & \Pi_3 \end{pmatrix} < 0 \tag{8}$$

where

$$\begin{aligned} \Pi_1 &= \text{diag}\{\Pi_{11}, \Pi_{22}, -Q_1, -Q_2, -(h_2 - h_1)^{-1} Z_1\} \\ \Pi_2^T &= \begin{pmatrix} PW_0 & PW_1 & PW_2 & PM & PM & PM & PM \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ \Pi_3 &= \text{diag}\{-\varepsilon_1 I, -\varepsilon_2 I, -\varepsilon_3 I, -\varepsilon_4 I, -\varepsilon_5 I, -\varepsilon_6 I, -\varepsilon_7 I\} \\ \Pi_{11} &= -PA - AP + \sum_{i=1}^3 Q_i + (h_2 - h_1) Z_1 + D_0 + \varepsilon_1 G^T G + \varepsilon_3 \tau^2 G^T G \\ &\quad + \varepsilon_4 N_1^T N_1 + \varepsilon_5 \lambda_{\max}(N_2^T N_2) G^T G + \varepsilon_7 \tau^2 \lambda_{\max}(N_4^T N_4) G^T G \\ \Pi_{22} &= (\mu - 1) Q_3 + D_1 + \varepsilon_2 G^T G + \varepsilon_6 \lambda_{\max}(N_3^T N_3) G^T G \end{aligned}$$

Proof. To obtain the result, the Lyapunov functional of system (1) is defined by

$$\begin{aligned}
 V(x_t, t) = & x^T(t)Px(t) + \sum_{i=1}^2 \int_{t-h_i}^t x^T(s)Q_i x(s)ds + \int_{t-d(t)}^t x^T(s)Q_3 x(s)ds \\
 & + \int_{-h_2}^{-h_1} \int_{t+\theta}^t x^T(s)Z_1 x(s)dsd\theta + \int_{-\tau}^0 \int_{t+\theta}^t x^T(s)Z_2 x(s)dsd\theta \quad (9)
 \end{aligned}$$

By Itô’s formula, we can calculate along the trajectories of the system (1), we have

$$\begin{aligned}
 \mathcal{L}V = & 2x^T(t)P[-(A + \Delta A)x(t) + (W_0 + \Delta W_0)f(x(t)) + (W_1 + \Delta W_1)f(y(t)) \\
 & + (W_2 + \Delta W_2) \int_{t-\tau}^t f(x(s))ds] + \text{trac}[\sigma^T(x(t), y(t), t)P\sigma(x(t), y(t), t)] \\
 & + x^T(t)Q_1 x(t) - x^T(t-h_1)Q_1 x(t-h_1) + x^T(t)Q_2 x(t) - x^T(t-h_2)Q_2 \\
 & x(t-h_2) + x^T(t)Q_3 x(t) - (1-d(\dot{t}))y^T(t)Q_3 y(t) + (h_2-h_1)x^T(t)Z_1 x(t) \\
 & - \int_{t-h_2}^{t-h_1} x^T(s)Z_1 x(s)ds + \tau x^T(t)Z_2 x(t)ds - \int_{t-\tau}^t x^T(s)Z_2 x(s) \quad (10)
 \end{aligned}$$

Here we note that, for positive $\varepsilon_i (i = 1 \dots, 7)$ it follows from Lemma 1 that

$$2x^T(t)PW_0 f(x(t)) \leq x^T(t)(\varepsilon_1 G^T G + \varepsilon_1^{-1}PW_0 W_0^T P)x(t) \quad (11)$$

$$2x^T(t)PW_1 f(y(t)) \leq \varepsilon_2 y^T(t)G^T G y(t) + \varepsilon_2^{-1}x^T(t)PW_1 W_1^T P x(t) \quad (12)$$

$$\begin{aligned}
 & 2x^T(t)PW_2 \int_{t-\tau}^t f(x(s))ds \\
 & \leq \varepsilon_3 \tau \int_{t-\tau}^t x^T(s)G^T G x(s)ds + \varepsilon_3^{-1}x^T(t)PW_2 W_2^T P x(t) \quad (13)
 \end{aligned}$$

$$-2x^T(t)P\Delta A x(t) \leq x^T(t)(\varepsilon_4 N_1^T N_1 + \varepsilon_4^{-1}PMM^T P)x(t) \quad (14)$$

$$2x^T(t)P\Delta W_0 f(x(t)) \leq x^T(t)(\varepsilon_5 \lambda_{max}(N_2^T N_2)G^T G + \varepsilon_5^{-1}PMM^T P)x(t) \quad (15)$$

$$\begin{aligned}
 & 2x^T(t)P\Delta W_1 f(y(t)) \\
 & \leq \varepsilon_6 \lambda_{max}(N_3^T N_3)y^T(t)G^T G y(t) + \varepsilon_6^{-1}x^T(t)PMM^T P x(t) \quad (16)
 \end{aligned}$$

$$\begin{aligned}
 & 2x^T(t)P\Delta W_2 \int_{t-\tau}^t f(x(s))ds \\
 & \leq \varepsilon_7 \tau \lambda_{max}(N_4^T N_4) \int_{t-\tau}^t x^T(s)G^T G x(s)ds + \varepsilon_7^{-1}x^T(t)PMM^T P x(t) \quad (17)
 \end{aligned}$$

Using (7), (11)-(17), and let $Z_2 = (\varepsilon_3 \tau + \varepsilon_7 \tau \lambda_{max}(N_4^T N_4))G^T G$, from (10) we have

$$\begin{aligned}
 \mathcal{L}V \leq & x^T(t) \left[-2PA + \sum_{i=1}^3 Q_i + (h_2-h_1)Z_1 + D_0 + \varepsilon_1 G^T G + \varepsilon_1^{-1}PW_0 W_0^T P \right. \\
 & \left. + \varepsilon_2^{-1}PW_1 W_1^T P + \varepsilon_3 \tau^2 G^T G + \varepsilon_3^{-1}PW_2 W_2^T P + (\varepsilon_4^{-1} + \varepsilon_5^{-1} + \varepsilon_6^{-1} + \varepsilon_7^{-1}) \right]
 \end{aligned}$$

$$\begin{aligned}
 &PMM^T P + \varepsilon_4 N_1^T N_1 + \varepsilon_5 \lambda_{max}(N_2^T N_2)G^T G + \varepsilon_7 \tau^2 \lambda_{max}(N_4^T N_4)G^T G \Big] x(t) \\
 &+ y^T(t) \left[(\mu - 1)Q_3 + D_1 + \varepsilon_2 G^T G + \varepsilon_6 \lambda_{max}(N_3^T N_3)G^T G \right] y(t) \\
 &- x(t - h_1)Q_1 x(t - h_1) - x(t - h_2)Q_2 x(t - h_2) \\
 &- (h_2 - h_1)^{-1} \left(\int_{t-h_2}^{t-h_1} x(s) ds \right)^T Z_1 \int_{t-h_2}^{t-h_1} x(s) ds
 \end{aligned} \tag{18}$$

Thus

$$\mathcal{L}V \leq \xi^T(t) \Pi^* \xi(t)$$

where

$$\begin{aligned}
 \Pi^* &= \text{diag}\{\Pi_{11}^*, \Pi_{22}^*, -Q_1, -Q_2, -(h_2 - h_1)^{-1} Z_1\} \\
 \Pi_{11}^* &= -PA - AP + \sum_{i=1}^3 Q_i + (h_2 - h_1)Z_1 + D_0 + \varepsilon_1 G^T G + \varepsilon_1^{-1} P W_0 W_0^T P \\
 &+ \varepsilon_2^{-1} P W_1 W_1^T P + \varepsilon_3 \tau^2 G^T G + \varepsilon_3^{-1} P W_2 W_2^T P + \sum_{j=4}^7 \varepsilon_j^{-1} P M M^T P + \varepsilon_4 N_1^T N_1 \\
 &+ \varepsilon_5 \lambda_{max}(N_2^T N_2)G^T G + \varepsilon_7 \tau^2 \lambda_{max}(N_4^T N_4)G^T G \\
 \Pi_{22}^* &= (\mu - 1)Q_3 + D_1 + \varepsilon_2 G^T G + \varepsilon_6 \lambda_{max}(N_3^T N_3)G^T G
 \end{aligned}$$

and $\xi^T(t) = \left[x^T(t) \quad y^T(t) \quad x^T(t - h_1) \quad x^T(t - h_2) \quad \left(\int_{t-h_2}^{t-h_1} x(s) ds \right)^T \right]$

From the Schur Complement Lemma, it is easy to know $\Pi^* < 0$ holds if and only if $\Pi < 0$. Hence for ensuring negativity of $\mathcal{L}V$ for any possible state, it suffices to require Π be a negative definite matrix. This implies that the system (1) is globally, asymptotically stable in the mean square. The proof is completed.

If $F = 0$, that is to say there are no uncertainty in stochastic neural networks. So Eq.(1) becomes:

$$\begin{aligned}
 dx(t) &= \left[-Ax(t) + W_0 f(x(t)) + W_1 f(y(t)) + W_2 \int_{t-\tau}^t f(x(s)) ds \right] dt \\
 &+ \sigma(x(t), y(t), t) dB(t)
 \end{aligned} \tag{19}$$

Then we have the following corollary

Corollary 1. Assume that there exist matrices $P > 0, D_0 > 0$ and $D_1 > 0$ such that

$$\text{trace}[\sigma^T(x(t), y(t), t) P \sigma(x(t), y(t), t)] \leq x^T(t) D_0 x(t) + y^T(t) D_1 y(t)$$

the stochastic neural network (19) is globally, asymptotically stable in the mean square, if there exist positive definite matrices Q_1, Q_2, Q_3, Z_1, Z_2 , and scalar $\varepsilon_1, \varepsilon_2, \varepsilon_3 > 0$ such that the LMI holds

$$\Xi = \begin{pmatrix} \Xi_{11} & 0 & 0 & 0 & 0 & PW_0 & PW_1 & PW_2 \\ \star & \Xi_{22} & 0 & 0 & 0 & 0 & 0 & 0 \\ \star & \star & -Q_1 & 0 & 0 & 0 & 0 & 0 \\ \star & \star & \star & -Q_2 & 0 & 0 & 0 & 0 \\ \star & \star & \star & \star & -(h_2 - h_1)^{-1}Z_1 & 0 & 0 & 0 \\ \star & \star & \star & \star & \star & -\varepsilon_1 I & 0 & 0 \\ \star & \star & \star & \star & \star & \star & -\varepsilon_2 I & 0 \\ \star & \star & \star & \star & \star & \star & \star & -\varepsilon_3 I \end{pmatrix} < 0 \quad (20)$$

where

$$\begin{aligned} \Xi_{11} &= -PA - AP + \sum_{i=1}^3 Q_i + (h_2 - h_1)Z_1 + \varepsilon_1 G^T G + \varepsilon_3 \tau^2 G^T G + D_0 \\ \Xi_{22} &= (\mu - 1)Q_3 + D_1 + \varepsilon_2 G^T G \end{aligned}$$

4 A Numerical Example

Consider a two-neuron stochastic neural network (1), where

$$\begin{aligned} A &= \begin{pmatrix} 6.5 & 0 \\ 0 & 5.5 \end{pmatrix}, W_0 = \begin{pmatrix} 1.5 & -1.6 \\ -1.6 & 1.5 \end{pmatrix}, W_1 = \begin{pmatrix} -0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}, W_2 = \begin{pmatrix} 1.2 & 0.3 \\ 0.3 & 0.9 \end{pmatrix}, \\ D_0 = D_1 &= \begin{pmatrix} 0.4 & 0 \\ 0 & 0.15 \end{pmatrix}, P = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}, M = \begin{pmatrix} 0.1 & 0.1 \\ 0 & 0.2 \end{pmatrix}, N_1 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}, \\ N_2 &= \begin{pmatrix} 0.3 & 0 \\ 0 & 0.2 \end{pmatrix}, N_3 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.2 \end{pmatrix}, N_4 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.3 \end{pmatrix}, \mu = 0.5, h_1 = 0.02, \\ h_2 &= 0.52, \tau = 0.2 \end{aligned}$$

$f_1(x) = f_2(x) = 0.5(|x + 1| - |x - 1|)$. We assume that $G = I, d_1(t) = d_2(t) = 0.5 \sin^2 t + 0.02$ and $\sigma(x(t), x(t-d(t)), t) = [0.2x_1(t) + 0.2x_1(t-d_1(t)) \quad 0.1x_2(t) + 0.1x_2(t-d_2(t))]^T$. Solving the LMI in Theorem 1, the feasible solution is obtained as

$$\begin{aligned} Q_1 = Q_2 &= \begin{pmatrix} 2.7297 & 2.5443 \\ 2.5443 & 3.1904 \end{pmatrix}, Q_3 = \begin{pmatrix} 5.4659 & 2.0960 \\ 2.0960 & 5.3587 \end{pmatrix}, Z_1 = \begin{pmatrix} 5.1100 & 4.7404 \\ 4.7404 & 5.9677 \end{pmatrix}, \\ Z_2 &= \begin{pmatrix} 4.6277 & 0 \\ 0 & 4.6277 \end{pmatrix}, \varepsilon_1 = 4.6516, \varepsilon_2 = 0.9789, \varepsilon_3 = 15.9691, \varepsilon_4 = 41.9973, \\ \varepsilon_5 &= 6.0939, \varepsilon_6 = 7.0486, \varepsilon_7 = 79.6593 \end{aligned}$$

Therefore, it follows from Theorem 1 that the two-neuron network (1) is robustly, globally, asymptotically stable in the mean square.

Acknowledgments. The project reported here was supported by the National Science Foundation of China with Grant Nos. 60874031 and 60740430664 and the Specialized Research Fund for the Doctoral Program of Higher Education of China 2007048750.

References

1. Arik, S.: Stability analysis of delayed neural networks. *IEEE Trans. Circuits Syst. I.* 47, 1089–1092 (2000)
2. Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: *Linear matrix inequalities in system and control theory.* SIAM, Philadelphia(PA) (1994)
3. Blythe, S., Mao, X., Liao, X.: Stability of stochastic delay neural networks. *J. Franklin Inst.* 338, 481–495 (2001)
4. Chen, T., Amari, S.: Stability of asymmetric Hopfield networks. *IEEE Trans. Neural Networks* 12, 159–163 (2001)
5. Feng, W., Yang, S., Fu, W., Wu, H.: Robust stability analysis of uncertain stochastic neural networks with interval time-varying delay. *Chaos Solitons Fract* (2008), doi:10.1016/j.chaos.2008.01.024
6. Gahinet, P., Nemirovsky, A., Laub, A.J., Chilali, M.: *LMI control toolbox: for use with Matlab works inc.* (1995)
7. Huang, H., Ho, D.W.C., Lam, J.: Stochastic stability analysis of fuzzy Hopfield neural networks with time-varying delays. *IEEE Trans. Circuits Syst-II* 52, 251–255 (2005)
8. He, Y., Liu, G., Rees, D., Wu, M.: Stability analysis for neural networks with time-varying interval delay. *IEEE Trans. Neural Networks* 18, 1850–1854 (2007)
9. He, Y., Wang, Q., Lin, C., Wu, M.: Delay-range-dependent stability for systems with time-varying delay. *Automatica* 43, 371–376 (2007)
10. He, Y., Wu, M., She, J.: Delay-dependent exponential stability for delayed neural networks with time-varying delay. *IEEE Trans. Circuits Syst-II* 53, 553–557 (2006)
11. Hu, J., Zhong, S., Liang, L.: Exponential stability analysis of stochastic delayed cellular neural network. *Chaos Solitons Fractals.* 27, 100–110 (2006)
12. Joya, G., Atencia, M.A., Sandoval, F.: Hopfield neural networks for optimization: study of the different dynamics. *Neurocomputing* 43, 219–237 (2002)
13. Li, W., Lee, T.: Hopfield neural networks for affine invariant matching. *IEEE Trans. Neural Networks* 12, 1400–1410 (2001)
14. Ruan, S., Filfil, R.S.: Dynamics of a two-neuron system with discrete and distributed delays. *Physica D.* 191, 323–342 (2004)
15. Wang, Z., Fang, J., Liu, X.: Global stability of stochastic high-order neural networks with discrete and distributed delays. *Chaos Solitons Fractals* 36, 388–396 (2008)
16. Wang, Z., Liu, Y., Liu, X.: On global asymptotic stability of neural networks with discrete and distributed delays. *Phys. Lett. A.* 345, 299–308 (2005)
17. Wang, Z., Qiao, H.: Robust filtering for bilinear uncertain stochastic discrete-time systems. *IEEE Trans. Signal Process* 50, 560–570 (2002)
18. Wan, L., Sun, J.: Mean square exponential stability of stochastic delayed Hopfield neural networks. *Phys. Lett. A.* 343, 306–318 (2005)
19. Wang, Z., Shu, H., Liu, Y., Ho, D.W., Liu, X.: Robust stability analysis of generalized neural networks with discrete and distributed time delays. *Chaos Solitons Fractals* 30, 886–896 (2006)
20. Young, S., Scott, P., Nasrabadi, N.: Object recognition using multi-layer Hopfield neural network. *IEEE Trans. Image Process* 6, 357–372 (1997)
21. Zhao, H.: Global asymptotic stability of Hopfield neural network involving distributed delays. *Neural Networks* 17, 47–53 (2004)
22. Zhang, Y., Heng, P.A., Leung, K.S.: Convergence analysis of cellular networks with unbounded delays. *IEEE Trans. Circuits Syst I.* 48, 680–687 (2001)

Hybrid Hopfield Architecture for Solving Nonlinear Programming Problems

Fabiana Cristina Bertoni and Ivan Nunes da Silva

State University of Feira de Santana, Department of Computer Engineering,
CEP 44031-460, Feira de Santana, BA, Brazil

University of São Paulo, Department of Electrical Engineering,
CP 359, CEP 13566.590, São Carlos, SP, Brazil

fcbertoni@gmail.com, insilva@sc.usp.br

<http://laips.sel.eesc.usp.br>

Abstract. This paper presents a neurogenetic approach for solving nonlinear programming problems. Genetic algorithm must its popularity to make possible cover nonlinear and extensive search spaces. Neural networks with feedback connections provide a computing model capable of solving a large class of optimization problems. The association of a modified Hopfield network with genetic algorithm guarantees the convergence of the system to the equilibrium points, which represent feasible solutions for nonlinear programming problems.

Keywords: Hopfield network, genetic algorithms, nonlinear programming.

1 Introduction

The nonlinear optimization plays a fundamental role in many problems involved with the areas of sciences and engineering, where a set of parameters is optimized subject to inequality and/or equality constraints [1].

In the neural networks literature, there exist several approaches used for solving constrained nonlinear optimization problems. The first neural approach applied to optimization problems was proposed by Tank and Hopfield in [2], where the network was used for solving linear programming problems. More recently, it was proposed in [3] a recurrent neural network for nonlinear optimization with lossy dynamics and time-varying activation functions. In [4] was developed a multilayer perceptron for nonlinear programming, which converts constrained optimization problems into a sequence of unconstrained ones by incorporating the constraint functions into the objective function of the unconstrained problem. In [5] was proposed a Hopfield neural network for constrained nonlinear optimization associated with Lagrange multipliers, which introduce the constraints into the objective function. The authors reported that the computation efficiency of the model is relatively low by using Lagrange multipliers, even so using parallel processing. In [6] was developed a new recurrent neural network for solving nonlinear optimization problems. The proposed neural network has

a one-layer structure and uses two penalty parameters that are experimentally obtained for each problem.

Basically, most of these neural networks presented in the literature for solving nonlinear optimization problems contain some penalty parameters. The stable equilibrium points of these networks, which represent a solution of the constrained optimization problems, are obtained only when those parameters are properly adjusted, and in this case, both the accuracy and the convergence process can be affected. In this paper, we propose a neurogenetic architecture for solving nonlinear programming problems.

2 The Modified Hopfield Neural Network

As introduced in [7], Hopfield networks are single-layer networks with feedback connections between nodes. In the standard case, the nodes are fully connected, i.e., every node is connected to all others nodes, including itself. The node equation for the continuous-time network with N -neurons is given by:

$$\dot{u}_i(t) = -\eta \cdot u_i(t) + \sum_{j=1}^N T_{ij} \cdot v_j(t) + i_i^b. \quad (1)$$

$$v_i(t) = g(u_i(t)). \quad (2)$$

where $u_i(t)$ is the current state of the i -th neuron, $v_j(t)$ is the output of the j -th neuron, i_i^b is the offset bias of the i -th neuron, $\eta \cdot u_i(t)$ is a passive decay term, and T_{ij} is the weight connecting the j -th neuron to i -th neuron.

In Equation (2), $g(u_i(t))$ is a monotonically increasing threshold function that limits the output of each neuron to ensure that network output always lies in or within a hypercube. It is shown in [7] that the network equilibrium points correspond to values $v(t)$ for which the energy function (3) associated with the network is minimized:

$$E(t) = -\frac{1}{2}v(t)^T \cdot T \cdot v(t) - v(t)^T \cdot i^b. \quad (3)$$

The mapping of nonlinear programming problems using a Hopfield network consists of determining the weight matrix T and the bias vector i^b to compute equilibrium points. A modified energy function $E^m(t)$ is used here, which is defined by:

$$E^m(t) = E^{conf}(t) + Min f(v). \quad (4)$$

where $E^{conf}(t)$ is a confinement term that groups all the constraints imposed by the problem, and $Min f(v)$ refers to the minimization of the objective function associated with the constrained optimization problem in analysis, which conducts the network output to the equilibrium points. Thus, the minimization of $E^m(t)$ of the modified Hopfield network is conducted in two stages:

i) Minimization of the Term $E^{conf}(t)$.

$$E^{conf}(t) = -\frac{1}{2}v(t)^T \cdot T^{conf} \cdot v(t) - v(t)^T \cdot i^{conf}. \tag{5}$$

where: $v(t)$ is the network output, T^{conf} is weight matrix and i^{conf} is bias vector belonging to E^{conf} . This corresponds to confinement of $v(t)$ into a valid subspace that confines the inequality constraints imposed by the problem. An investigation associating the equilibrium points with respect to the eigenvalues and eigenvectors of the matrix T^{conf} shows that all feasible solutions can be grouped in a unique subspace of solutions. A detailed description of this technique can be found in [8,9].

ii) Minimization of the Objective Function $f(v)$. After confinement of all feasible solutions to the valid subspace, a Genetic Algorithm (GA) is applied in order to optimize the objective function by inserting the values $v(t)$ into the chromosomes population. The operation of this hybrid system consists of three main steps as shown in Fig. 1:

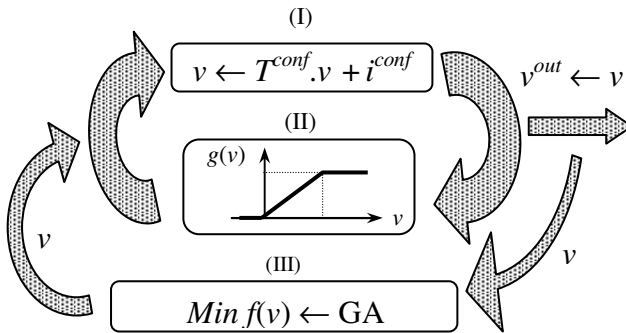


Fig. 1. The neurogenetic architecture for solving nonlinear programming

Step (I): Minimization of E^{conf} , corresponding to the projection of $v(t)$ in the valid subspace defined by:

$$v(t + 1) = T^{conf} \cdot v(t) + i^{conf}. \tag{6}$$

where: T^{conf} is a projection matrix ($T^{conf} \cdot T^{conf} = T^{conf}$ and $(T^{conf} \cdot i^{conf} = 0)$). This operation corresponds to an indirect minimization process of $E^{conf}(t)$ using orthogonal projection de $v(t)$ on the feasible set.

Step (II): Application of a *symmetric-ramp* activation function constraining $v(t)$ in a hypercube, i.e.

$$g(v_i) = \begin{cases} \lim_i^{inf} & , \text{ if } v_i < \lim_i^{inf} \\ v_i & , \text{ if } \lim_i^{inf} \leq v_i \leq \lim_i^{sup} \\ \lim_i^{sup} & , \text{ if } v_i > \lim_i^{sup} \end{cases} \quad (7)$$

where $v_i(t) \in [\lim_i^{inf}, \lim_i^{sup}]$.

Step (III): Minimization of $f(v)$, which involves the application of a genetic algorithm to move $v(t)$ towards an optimal solution that corresponds to network equilibrium points, which are the solutions for the constrained optimization problem considered.

As seen in Fig. 1, each iteration represented by the above steps has two distinct stages. First, as described in Step (III), v is updated using the genetic algorithm. Second, after each updating given in Step (III), v is projected directly in the valid subspace by the modified Hopfield network. This second stage is an iterative process, in which v is first orthogonally projected in the valid subspace by applying Step (I) and then thresholded in Step (II) so that its elements lie in the range defined by $[\lim_i^{inf}, \lim_i^{sup}]$. The convergence process is concluded when the values of v^{out} during two successive loops remain practically constant, where the value of v^{out} in this case is equal to v .

3 Genetic Algorithm for Objective Function Optimization

The algorithm begins by randomly developing the first population, where each individual is a possible solution for the problem. From this point, the fitness value in relation to each individual is computed. Based on this value, the elements that will belong to the next generation are selected (by election based on probabilistic criteria). To complete the population, the selected parents are reproduced through the implementation of genetic operators (crossing and mutation).

Codification: In this stage, the chromosomes $C_i = (c_{i1}, c_{i2}, \dots, c_{im})$ are encoded into sequences of binary digits and have a fixed size m , which represents the number of bits necessary to codification of a real number into the interval $[\lim_i^{inf}, \lim_i^{sup}]$. In our simulations, the value of m was assumed as 16.

Population Size: The population size used here was 100 individuals, which allowed for a better coverage of the search space and was efficient in our experiments.

Initial Population: The initial population is generated by introducing a chromosome that represents the values $v(t)$ previously obtained from Steps (I) and (II) described in Section 2. The remaining chromosomes are generated randomly.

Number of Generations: As stop criterion, it is verified the variation of the best individual from a generation to another one, and when there is no variation, the algorithm must finish its execution. Associated to this criterion, a maximum

number of 100 generations was also established, being enough for reach the minimum value to the objective function of a nonlinear programming problem.

Fitness Function: The fitness function for constrained optimization problems is the own objective function to be minimized. The most adapted individual will have the minimum fitness value.

Intermediate Population: Given a population in which each individual has received a fitness value, there are several methods to select the individuals upon whom the genetic algorithms of crossing and mutation will be applied. The selected individuals will make up a population called intermediate population. The selection method used here to separate the intermediate population was the roulette method [10] and the crossing and mutation rates were defined, respectively, at 70% and 1%, as recommended in the literature [10]. An elitism percentage of 10% was also used.

4 Formulation of the Nonlinear Programming Problem

Consider the following constrained optimization problem, with m -constraints and n -variables, given by the following equations:

$$\text{Minimize } f(v). \tag{8}$$

$$\text{subject to: } h_i(v) \leq 0, \quad i \in \{1..m\}. \tag{9}$$

$$z^{min} \leq v \leq z^{max}. \tag{10}$$

where $v, z^{min}, z^{max} \in \mathbb{R}^n$, $f(v)$ and $h_i(v)$ are continuous, and all first and second order partial derivatives of $f(v)$ and $h_i(v)$ exist and are continuous. The vectors z^{min} and z^{max} define the bounds on the variables belonging to the vector v . The parameters T^{conf} and i^{conf} are calculated by transforming the inequality constraints in (9) into equality constraints by introducing a slack variable $w \in \mathbb{R}^N$ for each inequality constraint:

$$h_i(v) + \sum_{j=1}^m q_{ij} \cdot w_j = 0. \tag{11}$$

where w_j are slack variables, treated as the variables v_i , and q_{ij} is defined by the Kronecker impulse function:

$$q_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases} \tag{12}$$

After this transformation, the problem defined by equations (8), (9) and (10) can be rewritten as:

$$\text{Minimize } f(v^+). \tag{13}$$

$$\text{subject to: } h^+(v^+) = 0. \tag{14}$$

$$z_i^{min} \leq v_i^+ \leq z_i^{max}, \quad i \in \{1..n\}. \tag{15}$$

$$0 \leq v_i^+ \leq z_i^{max}, \quad i \in \{(n + 1)..N\}. \tag{16}$$

where $N = n + m$, and $v^{+T} = [v^T \ w^T] \in \mathbb{R}^N$ is a vector of extended variables. Note that $f(v)$ does not depend on the slack variables w . In [11] has been shown that a projection matrix to the system (9) is given by:

$$T^{conf} = I - \nabla h(v^+)^T \cdot (\nabla h(v^+) \cdot \nabla h(v^+)^T)^{-1} \cdot \nabla h(v^+). \tag{17}$$

where:

$$\nabla h(v^+) = \begin{bmatrix} \frac{\partial h_1(v^+)}{\partial v_1^+} & \frac{\partial h_1(v^+)}{\partial v_2^+} & \dots & \frac{\partial h_1(v^+)}{\partial v_N^+} \\ \frac{\partial h_2(v^+)}{\partial v_1^+} & \frac{\partial h_2(v^+)}{\partial v_2^+} & \dots & \frac{\partial h_2(v^+)}{\partial v_N^+} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_m(v^+)}{\partial v_1^+} & \frac{\partial h_m(v^+)}{\partial v_2^+} & \dots & \frac{\partial h_m(v^+)}{\partial v_N^+} \end{bmatrix}. \tag{18}$$

Inserting the value of (17) in the expression of the valid subspace in (6), we have:

$$v^+ \leftarrow [I - \nabla h(v^+)^T \cdot (\nabla h(v^+) \cdot \nabla h(v^+)^T)^{-1} \cdot \nabla h(v^+)] \cdot v^+ + i^{conf}. \tag{19}$$

Results of the Lyapunov stability theory [12] should be used in (19) to guarantee the stability of the nonlinear system, and consequently, to force the network convergence to equilibrium points that represent a feasible solution to the nonlinear system. By the definition of the Jacobean, when v leads to equilibrium point implicates in $v^e = 0$. In this case, the value of i^{conf} should also be null to satisfy the equilibrium condition, i.e., $v^e = v(t) = v(t + 1) = 0$. Thus, $h(v^+)$ given in equation (19) can be approximated as follows:

$$h(v^+) \approx h(v^e) + J \cdot (v^+ - v^e). \tag{20}$$

where $J = \nabla h(v^+)$ and $h(v^+) = [h_1(v^+) \ h_2(v^+) \ \dots \ h_m(v^+)]^T$.

In the proximity of the equilibrium point $v^e = 0$, we obtain the following equation related to the parameters v^+ and $h(v^+)$:

$$\lim_{v^+ \rightarrow v^e} \frac{\|h(v^+)\|}{\|v^+\|} = 0. \tag{21}$$

Finally, introducing (20) and (21) in equation given by (19), we obtain

$$v^+ \leftarrow v^+ - \nabla h(v^+)^T \cdot (\nabla h(v^+) \cdot \nabla h(v^+)^T)^{-1} \cdot \nabla h(v^+). \tag{22}$$

Therefore, equation (22) synthesizes the valid-subspace expression for treating systems of nonlinear equations. In this case, the valid-subspace equation given in (6), which is represented by Step (I) in Fig. 1, should be substituted by equation (22).

5 Simulation Results

Problem 1. Consider the following constrained optimization problem, which is composed by inequality and equality constraints:

$$\text{Min } f(v) = v_1^3 + 2 \cdot v_2^2 \cdot v_3 + 2 \cdot v_3. \tag{23}$$

$$\text{subject to: } v_1^2 + v_2 + v_3^2 = 4. \tag{24}$$

$$v_1^2 - v_2 + 2 \cdot v_3^2 \leq 2. \tag{25}$$

$$v_1, v_2, v_3 \geq 0. \tag{26}$$

The optimal solution for this problem is given by $v^* = [0.00 \ 4.00 \ 0.00]^T$, where the minimal value of $f(v^*)$ at this point is equal to 0. Figure 2(a) shows the trajectories of the system variables starting from the initial point $v^0 = [1.67 \ 1.18 \ 3.37]^T$.

The trajectory of the objective function starting from initial point presented above is illustrated in Fig. 2(b). The system has also been evaluated for different values of initial conditions. All simulation results obtained by the neurogenetic system show that the proposed architecture is globally asymptotically stable at v^* .

Problem 2. Consider the following constrained optimization problem, which is also composed by inequality and equality constraints:

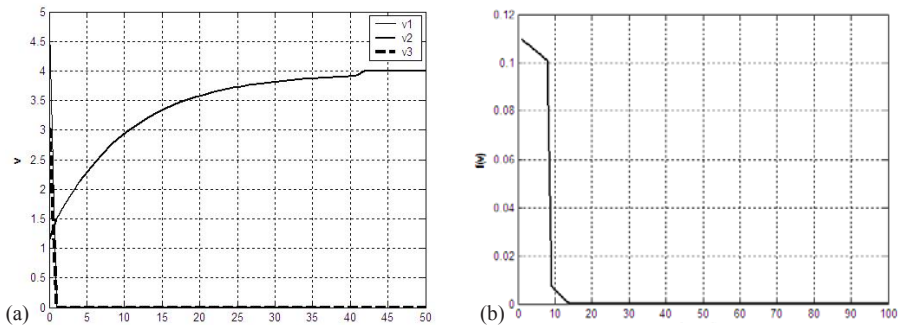


Fig. 2. Neurogenetic system output evolution (a), and objective function behavior (b)

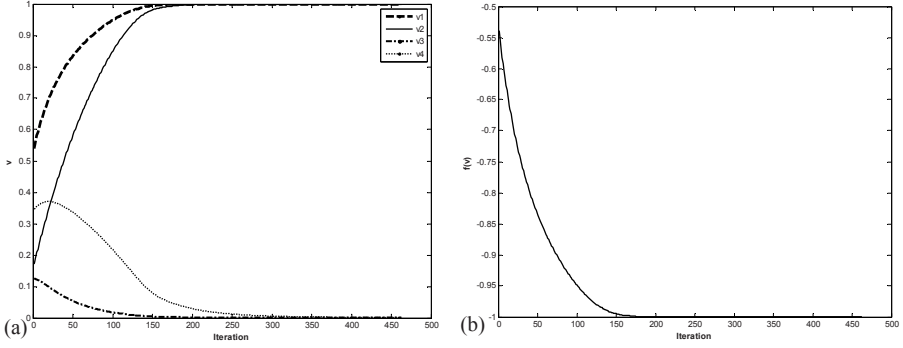


Fig. 3. Neurogenetic system output evolution (a), and objective function behavior (b)

$$\text{Min } f(v) = -v_1. \tag{27}$$

$$\text{subject to: } v_2 - v_1^3 - v_3^2 = 0. \tag{28}$$

$$v_1^2 - v_2 + v_4^2 = 0. \tag{29}$$

$$v_2 - v_1^3 \geq 0. \tag{30}$$

$$v_1^2 - v_2 \geq 0. \tag{31}$$

The optimal solution for this problem is $v^* = [1.00 \ 1.00 \ 0.00 \ 0.00]^T$ and the minimal value of $f(v^*)$ equal to 1.00. Figure 3 presents the evolution of the neurogenetic architecture output and objective function behavior, respectively.

The system has also been evaluated for different values of initial conditions. All simulation results show that this architecture is globally asymptotically stable at v^* .

Problem 3. Consider the following constrained optimization problem, which is just composed by inequality constraints:

$$\text{Min } f(v) = v_1^2 + 2 \cdot v_2^2 - 2 \cdot v_1 \cdot v_2 - 2 \cdot v_1 - 6 \cdot v_2 \tag{32}$$

$$\text{subject to: } v_1 + v_2 \leq 2. \tag{33}$$

$$-v_1 + 2 \cdot v_2 \leq 2. \tag{34}$$

This problem has an optimal solution $v^* = [0.8 \ 1.2]^T$ and $f(v^*) = 7.2$. Figure 4 presents the evolution of the neurogenetic architecture output and objective function behavior, respectively.

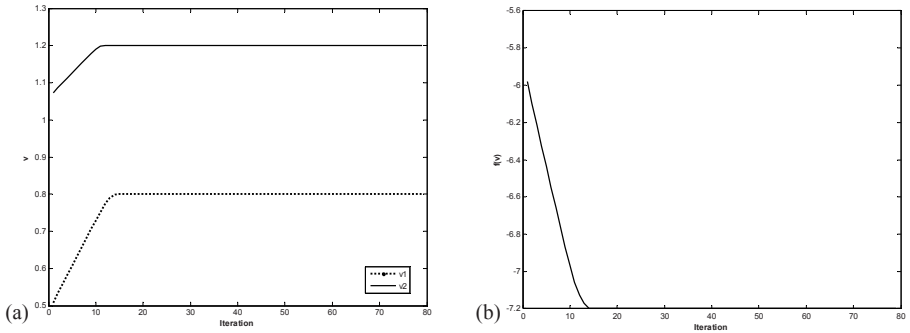


Fig. 4. Neurogenetic system output evolution (a), and objective function behavior (b)

The system has also been evaluated for different values of initial conditions. All simulation results show that this architecture is globally asymptotically stable at v^* .

6 Conclusions

This paper presented a neurogenetic approach for solving nonlinear programming problems. In contrast to the other neural approaches used in these types of problems, the main advantages of using the proposed approach in nonlinear optimization are the following: i) consideration of optimization and constraint terms in distinct stages with no interference with each other, i.e., the modified Hopfield network performs the optimization of constraints and the genetic algorithm is responsible for minimizing the objective function, ii) unique energy term (E^{conf}) to group all constraints imposed on the problem, iii) the internal parameters of the modified Hopfield network are explicitly obtained by the valid-subspace technique of solutions, which avoids the need to use training algorithm for their adjustments, and iv) optimization and confinement terms are not weighted by penalty parameters.

Some particularities of this neurogenetic approach in relation to primal methods normally used in nonlinear optimization are the following: i) it is not necessary the computation, in each iteration, of the active set of constraints; ii) the initial solution used to initialize the network can be outside of the feasible set defined from the constraints. The simulation results demonstrate that the neurogenetic system is an alternative method to solve nonlinear programming problems efficiently.

References

1. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming. Wiley, NY (1993); Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (1999)

2. Tank, D.W., Hopfield, J.J.: Simple Neural Optimization Networks: An A/D Converter, Signal Decision Network, and a Linear Programming Circuit. *IEEE Trans. on Circuits and Systems*. 33, 533–541 (1986)
3. Biro, J.J., Heszberger, Z.: An Optimization Neural Network Model with Lossy Dynamics and Time-Varying Activation Functions. In: *International Joint Conference on Neural Networks*, vol. 3, pp. 2245–2249 (2004)
4. Reifman, J., Feldman, E.E.: Multilayer Perceptron for Nonlinear Programming. *Computers & Operations Research* 29, 1237–1250 (2002)
5. Hao, X., Gao, H., Sun, C., Liu, B.: A Model Solving Constrained Optimization Problem Based on the Stability of Hopfield Neural Network. *Sixth World Congress on Intelligent Control and Automation* 1, 2790–2795 (2006)
6. Xia, Y., Feng, G.: A New Neural Network for Solving Nonlinear Projection Equations. *Neural Networks* 20, 577–589 (2007)
7. Hopfield, J.J.: Neurons With a Graded Response Have Collective Computational Properties Like Those of Two-State Neurons. *Proc. of the Nat. Acad. of Science*. 81, 3088–3092 (1984)
8. Aiyer, S.V., Niranjan, M., Fallside, F.: A Theoretical Investigation into the Performance of the Hopfield Network. *IEEE Trans. on Neural Networks*. 1, 53–60 (1990)
9. Da Silva, I.N., Amaral, W.C., Arruda, L.V.R.: Neural Approach for Solving Several Types of Optimization Problems. *Journal of Optimization Theory and Applications* 128, 563–580 (2006)
10. Mitchell, M.: *An Introduction to Genetic Algorithms*. MIT Press, Cambridge (1996)
11. Luenberger, D.G.: *Linear and Nonlinear Programming*. Springer, New York (2003)
12. Vidyasagar, M.: *Nonlinear Systems Analysis*. Prentice-Hall, Englewood Cliffs (1993)

Fault Tolerant Regularizers for Multilayer Feedforward Networks

Deng-yu Qiao¹, Chi Sing Leung¹, and Pui Fai Sum²

¹ Dept. of Electronic Engineering, City University of Hong Kong, Hong Kong

² Institute of Electronic Commerce, National Chung Hsing University

Abstract. In multilayer feedforward networks (MFNs), when open weight fault exists, many potential faulty networks should be considered during training. Hence, the objective function, as well as the corresponding learning algorithm, would be computationally complicated. This paper derives an objective function for improving the fault tolerance of MFNs. With the linearization technique, the objective function is decomposed into two terms, the training error and a simple regularization term. In our approach, the objective function is computational simple. Hence, the conventional backpropagation algorithm can be simply applied to handle this fault tolerant objective function. Simulation results show that compared with the conventional approach, our approach has a better fault tolerant ability.

1 Introduction

The implementation of neural networks on physical hardware cannot be perfect [1,3]. If special training methods are not taken, the fault situation could lead to a drastic performance degradation. Therefore, we would like to have a trained network with an ability to handle network faults. One of important fault models is the open weight fault [2,4,5]. In this fault model, some connected weight are disconnected. There were several algorithms for handling this fault model.

One technique is to limit the weight magnitude [6,7]. One deficiency of limiting weight magnitude is that the theoretical justification on the underlying objective function is not so clear. In [6], the way to set the weight decay constant is not discussed even the fault statistics is available. Besides, we can also formulate the training process as solving a minmax problem [8,9]. Their drawback is that solving a minmax problem is very complicated. Zhou *et al.* [4] defined a similar objective function and developed the corresponding learning algorithm. In the Zhou's approach, the objective function consists of two terms. One is term is the training error of a fault-free network. Another term is the sum of the training errors of some potential faulty networks. In [4], Zhou *et al.* empirically showed that the proposed objective function can improve generalization and fault tolerance.

The above formulations are effective to handle small networks and single weight fault. However, when the network size is large and the multi-weight fault appears, the number of possible faulty network is very large. For example, in a

multilayer feedforward network (MFN) [10,11] with M weights, for multi-weight fault, the number of potential faulty networks is $N_{\text{networks}} = \sum_{i=1}^n C_n^M$, where n is the maximum number of faulty weights that we considered. Also, in [4], the theoretical guideline to set the weighting factors was not addressed.

This paper develops an objective function for multilayer feedforward networks (MFNs) for multi-weight open fault. We use the average error to build the objective function. Afterwards, with the linearization on the objective function, a regularizer is identified in the objective function. With the proposed objective function, we can develop gradient based learning methods, such as the standard backpropagation [10] and fast backpropagation [11], for the fault tolerant objective function.

The organization of this paper is as follows. In Section 2, we review the concept of BP networks and fault tolerance. In Section 3, the objective function for multi-weight open fault and the corresponding regularizer are defined. Section 4 presents our simulation results. Conclusion is presented in Section 5.

2 Background

In this paper, we assume that the training data set

$$\mathcal{D}_T = \{(\mathbf{x}_i, y_i) : \mathbf{x}_i \in \mathcal{R}^K, y_i \in \mathcal{R}, i = 1, \dots, N.\}, \quad (1)$$

is generated by an unknown stochastic system [12] [13], given by

$$y_i = f(\mathbf{x}_i) + e_i \quad (2)$$

where \mathbf{x}_i and y_i are the input and output samples of the system system, respectively, K is the input dimension, $f(\cdot)$ is the unknown system, and e_i 's are the random measurement noise. The noise e_i 's are independent zero-mean Gaussian random variables with variance equal to σ_e^2 . Now, our problem is to construct a model for approximating the mapping $f(\cdot)$.

In the MFN approach with M weights, we would like to approximate the mapping $f(\cdot)$ by an MFN, given by

$$f(\mathbf{x}) \approx h(\mathbf{x}, \mathbf{w}) \quad (3)$$

where $h(\mathbf{x}, \mathbf{w})$ is the network output function, and \mathbf{w} is the weight vector that contains all M weights in the MFN. The classical training objective is the training error, given by

$$\mathcal{E}_t(\mathbf{w}) = \frac{1}{N} \sum_i^N (y_i - h(\mathbf{x}_i, \mathbf{w}))^2. \quad (4)$$

In the multi-weight open fault model, the faulty weight vector is described by a weight multiplicative model, given by

$$\tilde{w}_j = b_j w_j, \quad (5)$$

where w_j is the j -th element in weight vector \mathbf{w} ; and the fault factor b_j describes whether the j -th weight operates properly or not. If $b_j = 0$, the j -th weight is out of work. Otherwise, the j -th weight operates properly. Define $\mathbf{b} = [b_1, \dots, b_M]^T$ as the fault vector. In vector-matrix notation, (5) can be rewritten as

$$\tilde{\mathbf{w}} = \mathbf{b} \otimes \mathbf{w} \tag{6}$$

where \otimes is the element-wise multiplication operator. For a particular fault vector \mathbf{b} , the training error is given by

$$\mathcal{E}_b(\mathbf{w}, \mathbf{b}) = \frac{1}{N} \sum_{i=1}^N (y_i - h(\mathbf{x}_i, \tilde{\mathbf{w}}))^2. \tag{7}$$

In [4], the following objective function is used

$$\mathcal{J}(\mathbf{w})_{\mathbf{z}} = \alpha \mathcal{E}_t(\mathbf{w}) + \beta \frac{1}{N_b} \sum_{\mathbf{b} \in \mathcal{S}_b} \mathcal{E}_b(\mathbf{w}, \mathbf{b}) \tag{8}$$

where \mathcal{S}_b is the set of all possible fault vectors considered, and N_b is the number of elements in \mathcal{S}_b . The parameters α and β are the weighting factors. In [4], it is shown that the training algorithm, based on (8), can improve fault tolerant as well as generalization. However, when the number of faulty weights is greater than one, the number of potential faulty networks is very large. Also, the rule to set the two weighting factors was not theoretically discussed [2,4].

3 Objective Function and Regularizer

Consider the linearization on the network output $h(\mathbf{x}_i, \mathbf{w})$ around $\hat{\mathbf{w}}$,

$$h(\mathbf{x}_i, \mathbf{w}) = \mathbf{H}_i^T \mathbf{w} + \xi_i \tag{9}$$

where

$$\mathbf{H}_i = \left. \frac{\partial h(\mathbf{x}_i, \mathbf{w})}{\partial \mathbf{w}} \right|_{\mathbf{w}=\hat{\mathbf{w}}}. \tag{10}$$

In the above, $\hat{\mathbf{w}}$ denotes the approximation value of \mathbf{w} in the last iteration, and ξ_i is the residual in the expansion of $h(\mathbf{x}_i, \mathbf{w})$ given by

$$\xi_i = h(\mathbf{x}_i, \hat{\mathbf{w}}) - \mathbf{H}_i^T \hat{\mathbf{w}} + \rho_i \tag{11}$$

where ρ_i is the higher order residual. Our learning task is to find out the weights that best fits the observations. Recall that for a faulty network, the average error is given by

$$\mathcal{E}_b(\mathbf{w}, \mathbf{b}) = \frac{1}{N} \sum_{i=1}^N (y_i - h(\mathbf{x}_i, \tilde{\mathbf{w}}))^2, \tag{12}$$

where $\tilde{\mathbf{w}} = \mathbf{b} \otimes \mathbf{w}$. Consider the linearization in (9), (12) can be rewritten as

$$\mathcal{E}_b(\mathbf{w}, \mathbf{b}) = \frac{1}{N} \sum_{i=1}^N (y_i - \mathbf{H}_i^T \tilde{\mathbf{w}} - \xi_i)^2. \quad (13)$$

Assume the fault rate is equal to p . The average of $\mathcal{E}_b(\mathbf{w}, \mathbf{b})$ over all fault vector is given by

$$\begin{aligned} \bar{\mathcal{E}}_b(\mathbf{w}, \mathbf{b}) = \frac{1}{N} \sum_{i=1}^N \left[(y_i - \xi_i)^2 - 2(y_i - \xi_i) \left\langle \sum_{j=1}^M b_j w_j H_i^j \right\rangle_{\mathcal{S}_b} \right. \\ \left. + \left\langle \left(\sum_{j=1}^M b_j w_j H_i^j \right)^2 \right\rangle_{\mathcal{S}_b} \right] \end{aligned} \quad (14)$$

where H_i^j denotes the j th element of \mathbf{H}_i , and \mathcal{S}_b denotes the set of all possible fault vectors when the fault rate is equal to p . Since $\langle b_j \rangle = \langle b_j^2 \rangle = 1 - p$ and $\langle b_j b_{j'} \rangle = (1 - p)^2$ for $j \neq j'$, we have

$$\begin{aligned} \left\langle \sum_{j=1}^M b_j w_j H_i^j \right\rangle_{\mathcal{S}_b} &= (1 - p) \sum_{j=1}^M w_j H_i^j \\ \frac{1}{N} \sum_{i=1}^N \left\langle \left(\sum_{j=1}^M b_j w_j H_i^j \right)^2 \right\rangle_{\mathcal{S}_b} &= \mathbf{w}^T ((1 - p)\mathbf{G} + (1 - p)^2(\mathbf{H}_\phi - \mathbf{G}))\mathbf{w}, \end{aligned}$$

where $\mathbf{H}_\phi = \frac{1}{N} \sum_{i=1}^N \mathbf{H}_i \mathbf{H}_i^T$ and $\mathbf{G} = \text{diag}(\mathbf{H}_\phi)$. Now, the objective function $\bar{\mathcal{E}}_t(\mathbf{w}, \mathbf{b})$ can be rewritten as

$$\begin{aligned} \bar{\mathcal{E}}_b(\mathbf{w}, \mathbf{b}) &= \frac{1}{N} \sum_{i=1}^N (y_i - \xi_i)^2 - 2(1 - p) \frac{1}{N} \sum_{i=1}^N (y_i - \xi_i) \mathbf{H}_i^T \mathbf{w} \\ &\quad + (1 - p) \mathbf{w}^T \{(1 - p)\mathbf{H}_\phi + p\mathbf{G}\} \mathbf{w}. \end{aligned} \quad (15)$$

Since the term $\frac{1}{N} \sum_{i=1}^N (y_i - \xi_i)^2$ in (15) is independent of \mathbf{w} , we can re-scale this term by a constant $(1 - p)$. Thus we have

$$\begin{aligned} \bar{\mathcal{E}}_b(\mathbf{w}, \mathbf{b}) &= \frac{1}{N} \sum_{i=1}^N (y_i - \mathbf{H}_i^T \mathbf{w} - \xi_i)^2 + \mathbf{w}^T \mathbf{R} \mathbf{w} \\ &= \frac{1}{N} \sum_{i=1}^N (y_i - h(\mathbf{x}_i, \mathbf{w}))^2 + \mathbf{w}^T \mathbf{R} \mathbf{w} \\ &= \mathcal{E}_t(\mathbf{w}) + \mathbf{w}^T \mathbf{R} \mathbf{w}, \end{aligned} \quad (16)$$

where

$$\mathbf{R} = p(\mathbf{G} - \mathbf{H}_\phi). \quad (17)$$

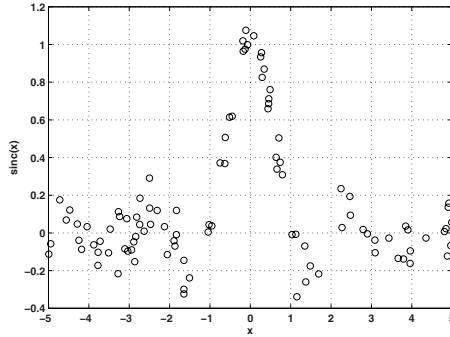


Fig. 1. Training data for the sinc function

In (16), the second term is similar to the conventional regularization term in regularization techniques. Hence, we could define the *multi-weight open fault regularizer* as

$$\mathbf{w}^T \mathbf{R} \mathbf{w}, \tag{18}$$

where \mathbf{R} is the so-called regularization matrix. The objective in (16) can be used to training the MFNs with the popular backpropagation rule [10, 11].

4 Simulation

4.1 Fault Tolerance

Sinc function. The sinc function is a common benchmark example [13, 14]. The output is generated by

$$y = \text{sinc}(x) + e, \tag{19}$$

where the noise term e is a mean zero Gaussian noise with variance $\sigma_e^2 = 0.01$. A training data set (100 samples), shown in Fig. 1, is generated. Also, a noise-free testing data set (100 samples) is generated. The MFN has one hidden layer with 25 hidden nodes. For each fault rate, we use the fast training algorithm [11] to train five MFNs with different initial weights. For each trained MFN, we randomly generate 10,000 faulty networks.

The training and testing MSEs are depicted in the Fig. 2. The standard BP method with the classical objective function gives out a very poor result on the training and test errors. The Zhou’s method and our robust method can greatly improve fault tolerance. For faulty networks ($p > 0$), among those three algorithms, our algorithm gives out the best fault tolerance.

Nonlinear time series example. We consider the following nonlinear autoregressive (NAR) time series [13], given by

$$y_i = (0.8 - 0.5 \exp(-y_{i-1}^2)) y_{i-1} - (0.3 + 0.9 \exp(-y_{i-1}^2)) y_{i-2} + 0.1 \sin(\pi y_{i-1}) + e_i \tag{20}$$

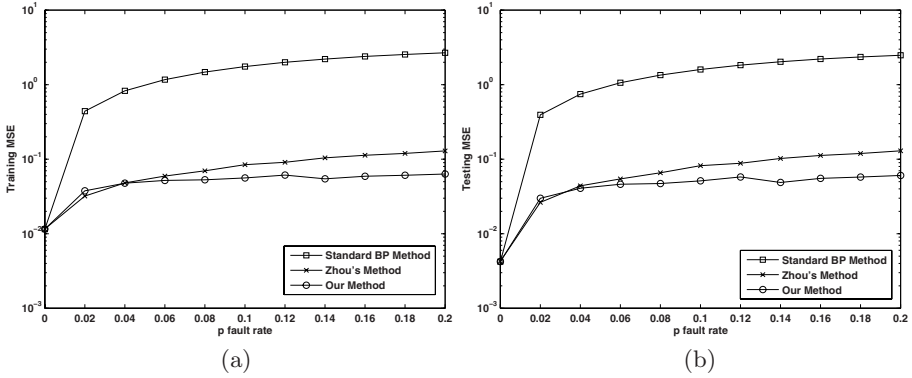


Fig. 2. MSEs of faulty networks for the noisy sinc function approximation, where $\sigma_e^2 = 0.01$. (a) Training error, (b) Testing error.

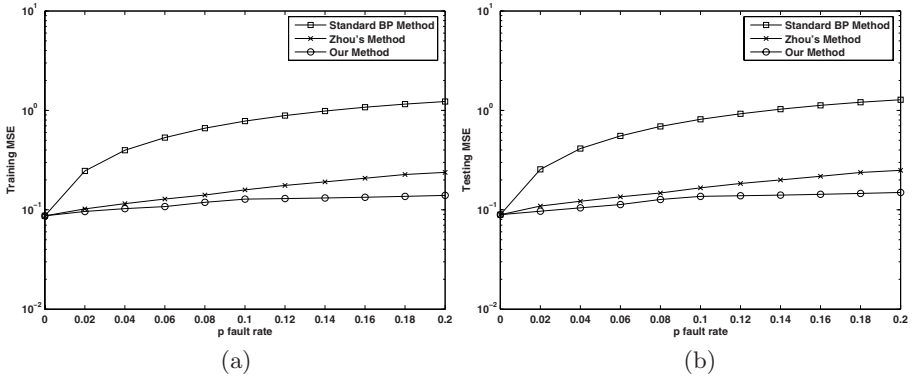


Fig. 3. MSEs of faulty networks for the NAR prediction. (a) Training error, (b) Testing error.

where e_i is a mean zero Gaussian random variable that drives the series. Its variance is equal to 0.09. One thousand samples were generated given $y_0 = y_{-1} = 0$. The first 500 data points, were used for training and the other 500 samples were used for testing. Our MFN model is used to predict y_i based on the past observations, y_{i-1} and y_{i-2} . The MFN model has one hidden layer with 25 hidden nodes. For each fault rate, we train five MFNs with different initial weights. For each trained MFN, we randomly generate 10,000 faulty networks. The performance of the trained BP networks are depicted in Fig. 3. Similar to the previous example, our algorithm gives out the best fault tolerance.

4.2 Incorrect Training Fault Rates

With our robust learning, we can optimize the weight vector with respect to the fault tolerance if we know the exact fault rate, i.e., fault statistics. In some

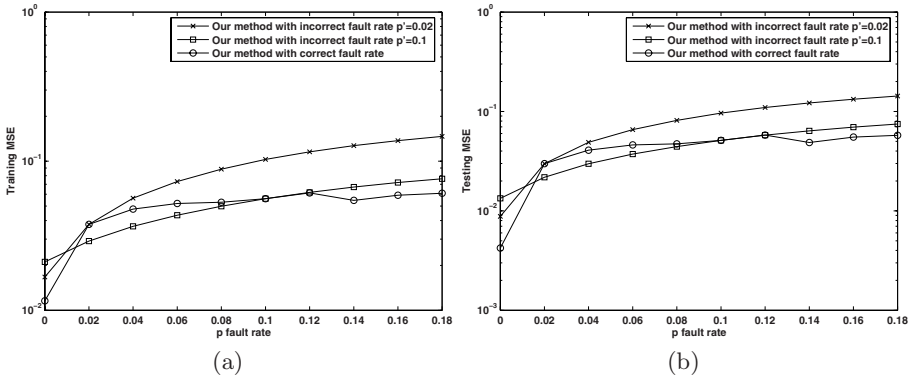


Fig. 4. MSEs of faulty networks with incorrect guess fault rate for the noisy sinc function example

practical situations, there may be a difference between the training fault rate and the exact fault rate. Hence, it is interesting to study the degradation due to this deviation.

The examples we considered here are the noisy sinc function with $\sigma_e = 0.01$. The setting of the MFN model used here is the same as before. In the simulation, for our robust method, we use two training fault rates, $p' = 0.02$ and 0.1 , to train BP networks. The MSEs of the BP networks trained with training fault rates are then measured on various true fault rates. For trained MFN, we randomly generate 60,000 faulty networks. The MSE performances are depicted in Fig 4.

For the training error, the degradation of our robust method due to using an incorrect training fault rate is not so large. For a small training fault rate $p' = 0.02$, the degradation becomes larger as the true fault rate increases. For a large training fault rate $p' = 0.1$, the degradation is large when the true fault rate is small. For the testing error, the result is quite similar to that of training error.

5 Conclusion

This paper addresses the fault tolerance of BP networks where all weights have the same fault rate and their fault probabilities are independent. We have derived an objective function for robustly training a BP network. Moreover, our method can handle the multi-weight open fault, compared with Zhou’s method [4]. Various simulation studies confirm that in terms of fault tolerance our approach is better than other methods being tested.

Acknowledgment

The work was supported by a research grant from City University of Hong Kong (7002480).

References

1. Burr, J.: Digital neural network implementations. In: *Neural Networks, Concepts, Applications, and Implementations*, vol. III. Prentice-Hall, Englewood Cliffs (1991)
2. Phatak, D.S., Koren, I.: Complete and partial fault tolerance of feedforward neural nets. *IEEE Trans. Neural Networks* 6, 446–456 (1995)
3. Murray, A.F., Edwards, P.J.: Enhanced mlp performance and fault tolerance resulting from synaptic weight noise during training. *IEEE Trans. Neural Networks* 5, 792–802 (1994)
4. Zhou, Z.H., Chen, S.F.: Evolving fault-tolerant neural networks. *Neural Computing and Applications* 11, 156–160 (2003)
5. Emmerson, M.D., Damper, R.I.: Determining and improving the fault tolerance of multilayer perceptrons in a pattern-recognition application. *IEEE Trans. Neural Networks* 4, 788–793 (1993)
6. Chiu, C.T., Mehrotra, K., Mohan, C.K., Ranka, S.: Modifying training algorithms for improved fault tolerance. In: *Proceedings of ICNN 1994*, vol. 4, pp. 333–338 (1994)
7. Cavaliere, S., Mirabella, O.: A novel learning algorithm which improves the partial fault tolerance of multilayer neural networks. *Neural Networks* 12, 91–106 (1999)
8. Neti, C., Schneider, M.H., Young, E.D.: Maximally fault tolerance neural networks. *IEEE Trans. Neural Networks* 3, 14–23 (1992)
9. Deodhare, D., Vidyasagar, M., Keerth, S.S.: Synthesis of fault-tolerant feedforward neural networks using minimax optimization. *IEEE Trans. Neural Networks* 9, 891–900 (1998)
10. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: *Parallel Distributed Processing I*. MIT Press, Cambridge
11. Vogl, T.P., Mangis, J.K., Rigler, A.K., Zink, W.T., Alkon, D.L.: Accelerating the convergence of the back-propagation methods. *Biological Cybernetics* 59(4-5), 257–263 (1988)
12. Amari, S.I., Murata, N., Muller, K.R., Finke, M., Yang, H.H.: Asymptotic statistical theory of overtraining and cross-validation. *IEEE Trans. Neural Networks* 8, 985–996 (1997)
13. Chen, S., Hong, X., Harris, C.J., Sharkey, P.M.: Sparse modelling using orthogonal forward regression with press statistic and regularization. *IEEE Trans. Systems, Man and Cybernetics, Part B*, 898–911 (2004)
14. Cherkassky, V., Ma, Y.: Multiple model regression estimation. *IEEE Trans. Neural Networks* 16(4), 785–798 (2005)

Integrating Simulated Annealing and Delta Technique for Constructing Optimal Prediction Intervals

Abbas Khosravi, Saeid Nahavandi, and Doug Creighton

Centre for Intelligent Systems Research (CISR)
Deakin University, Geelong, Australia
{akhos, saeid.nahavandi, doug.creighton}@deakin.edu.au

Abstract. This paper aims at developing a new criterion for quantitative assessment of prediction intervals. The proposed criterion is developed based on both key measures related to quality of prediction intervals: length and coverage probability. This criterion is applied as a cost function for optimizing prediction intervals constructed using delta technique for neural network model. Optimization seeks out to minimize length of prediction intervals without compromising their coverage probability. Simulated Annealing method is employed for readjusting neural network parameters for minimization of the new cost function. To further ameliorate search efficiency of the optimization method, parameters of the network trained using weight decay method are considered as the initial set in Simulated Annealing algorithm. Implementation of the proposed method for a real world case study shows length and coverage probability of constructed prediction intervals are better than those constructed using traditional techniques.

Keywords: prediction interval, neural network, simulated annealing, delta technique.

1 Introduction

Neural Networks (NNs) have achieved great success on many regression and classification problems in the last two decades. No matter how NNs are trained or used, they suffer from some basic deficiencies. One of the biggest concerns about NNs is how well they do point prediction task under presence of uncertainty. Source of uncertainty can be in data (e.g., measurement noise or some missing data), or in operation of the underlying system (e.g., occurrence of probabilistic events in complex systems). Although both of these types of uncertainties are common, the second type has more severe impacts on targets. It often leads to multiple realities for future of a system even under fixed conditions. For instance, in manufacturing enterprises, probable failure of some machines can originate formation of long queues before bottlenecks. This condition will significantly increase the lead time of products. Because NNs only generate a conditional mean of the training samples, their point prediction error for stochastic systems will be always big (regardless of NN type or size). The second problem of NN models for point prediction is lack of a measure about their estimation accuracy. In literature, often smallness of an error-based measure like Mean Squared Error (MSE)

or Mean Absolute Percentage Error (MAPE) computed for training or validation sets is claimed as an indication of reliability of NN models. Unfortunately, those measures carry no information about accuracy of point prediction and its reliability for unseen observations.

These two deficiencies have encouraged many researchers to construct Prediction Intervals (PIs) for outputs of NN models. Because different sources of uncertainty are covered in construction of PIs, they are practically more useful than confidence intervals and more reliable than point prediction. Mathematically, a prediction interval with confidence level of $(1 - \alpha)\%$ is a random interval based on past observations $\mathbf{x} = (x_1, x_2, \dots, x_n)$ and built for unseen observations X_{n+h} , $h \geq 1$,

$$I(h, \mathbf{x}) = [L(\mathbf{x}), U(\mathbf{x})] \quad (1)$$

so that $P(L(\mathbf{x}) \leq X_{n+h} \leq U(\mathbf{x})) = 1 - \alpha$. Without construction of prediction intervals, the validity of decisions made based on point prediction is always questionable.

In literature, a variety of techniques has been proposed and examined for PI construction. A good review of Bayesian, bootstrap, and delta techniques can be read in [1]. Although implementation of bootstrap technique has become feasible largely due to availability of powerful computers, it suffers from a high computational burden. Work done by Hwang et al. [2] showed that asymptotically valid PIs for NNs can be constructed based on theories of nonlinear regression. To avoid problems related to NN over-fitting, De Veaux et al. [3] developed a PI construction technique based on regularization and weight decay. Bayesian technique requires calculation of the Hessian matrix which makes it computationally expensive [4]. Application of PIs constructed using NNs has proliferated in recent years and many have used them instead of point predictions in different fields, among others, including temperature prediction [5], boring process prediction [6], paper curl prediction [7], modeling of solder paste deposition process [8], and time series forecasting [9].

Despite all these attempts, construction of PIs for NN models and their assessment still require more attention. The main focus of literature is on construction of PIs. Often, there is no discussion about assessment of constructed PIs in terms of both their length and coverage probability [6] [7] [8]. Very often only coverage probability and upper and lower bounds of PIs are stated, without any discussion on how wide or reliable these intervals are. Furthermore, the research and practice literature is void of papers about how PI construction techniques can be optimized. The purpose of such an optimization can be minimization of length of PIs without compromising their coverage probability.

Motivated by these gaps in literature, this study first attempts to develop a practically useful measure for quantitative evaluation of PIs in term of their length and coverage probability. This measure is mainly composed of indices about how well PIs cover the underlying targets and how wide they are compared with the range of targets. Secondly, a new cost function is developed based on these measures and is minimized in order to find the optimum values of some critical parameters of NN models. Through a case study, it is shown that constructed PIs based on the proposed optimization technique, will yield narrower PIs with the same nominal coverage probability (confidence level)

The rest of this paper is organized as follows: Section 2 briefly describes background required for this study. The assessment measure is introduced in Section 3.

The optimization procedure is discussed in Section 4. Section 5 represents numerical experiments conducted in this research and reports the results. Finally, the paper concludes with our observations and plans for future work.

2 Background

2.1 Delta Technique

Because this research is about PIs constructed using the delta technique, here we briefly introduce this technique. Its mathematical discussion and fundamental theories can be found in [2] and [3].

The original delta technique is based on Taylor series expansion of NN model around its optimal parameters, determined by minimization of Residual Sum of Squares (RSS). The error terms associated with the modeling function are assumed to be independently and identically distributed (iid) with variance δ^2 . According to theories of nonlinear regression, a $(1 - \alpha)\%$ asymptotic prediction interval for \hat{y}_i will be as follows,

$$\hat{y}_i \pm t_d^{1-\frac{\alpha}{2}} s [1 + \nabla_{w^T y}^T (J^T)^{-1} \nabla_{w y}]^{\frac{1}{2}} \quad i = 1, 2, \dots, m \tag{2}$$

where J is the Jacobian matrix of NN, w is the set of network parameters, s is the unbiased estimate of δ^2 , and $t_d^{1-\frac{\alpha}{2}}$ is $1 - \frac{\alpha}{2}$ quantile of a cumulative t-distribution function with d degrees of freedom. In case of using a weight decay regularizer to avoid over-fitting problem ($RSS + \lambda w^T w$), the PIs will be constructed as follows [3],

$$\hat{y}_i \pm t_d^{1-\frac{\alpha}{2}} s [1 + \nabla_{w^T y}^T (J^T + \lambda I)^{-1} (J^T) (J^T + \lambda I)^{-1} \nabla_{w y}]^{\frac{1}{2}} \quad i = 1, 2, \dots, m. \tag{3}$$

Construction of PIs based on (3) appropriately solves problems related to singularity of (J^T) in (2). Therefore, we will use this equation for construction of PIs, as it has been recommended in literature [10].

2.2 Simulated Annealing

The Simulated Annealing (SA) is a Monte Carlo technique that can be used for seeking out the global minimum. The effectiveness of SA is attributed to the nature that it can explore the design space by means of neighborhood structure and escape from local minima by probabilistically allowing uphill moves. Compared with traditional mathematical optimization techniques, SA offers a number of advantages: first, it is not derivative based, which means that it can be used for optimization of any cost function, regardless of its complexity or dimensionality, and secondly, it can explore and exploit the parameter space without being trapped in local optima (minima). SA has been shown to perform well for optimizing a wide variety of complex problems [11]. More information about SA and its optimization procedure can be found in [12] and [13].

3 The Proposed Assessment Measure for PIs

PIs can be characterized based on their length and coverage probability. One approach for quantitative assessment of PI lengths is to normalize each interval length with regard to range of targets. Following this, a measure called Normalized Mean Prediction Interval Length (NMPIL) can be obtained as follows:

$$NMPIL = \frac{1}{m} \sum_{i=1}^m \frac{U(x_i) - L(x_i)}{t_{\max} - t_{\min}} \tag{4}$$

where $U(x_i)$ and $L(x_i)$ are upper bound of PI, lower bound of PI. t_{\max} and t_{\min} are also extreme values of targets. Normalization of PI length by the range of targets makes objective comparison of PIs possible, regardless of techniques used for their construction or magnitudes of the underlying targets.

The PI Coverage Probability (PICP) indicates the probability that the underlying target will lie within the constructed PIs. It can be calculated through counting covered targets by PIs:

$$PICP = \frac{1}{m} \sum_{i=1}^m c_i \tag{5}$$

where,

$$c_i = \begin{cases} 1 & y_i \in [L(x_i), U(x_i)] \\ 0 & y_i \notin [L(x_i), U(x_i)] \end{cases} \tag{6}$$

It is always desirable to construct PIs whose PICP is the highest possible value. Such high PICP can be simply achieved through considering target ranges as PIs for all samples. Needless to say, wide PIs like these ones are practically useless. This argument makes clear that judgment about PIs based on PICP without considering length of PIs is always subjective and biased. It is essential to evaluate PIs simultaneously based on their both key measures: length and coverage probability.

Generally, PI lengths and PICP have a direct relationship. The wider the PIs, the higher the corresponding PICP. This means that as soon as PIs are squeezed, some targets will lie out of PIs, which results in low PICP. According to this discussion, the following Coverage-Length-based Criterion (CLC) is proposed for comprehensive evaluation of PIs in term of their PICP and lengths:

$$CLC = \frac{NMPIL}{\sigma(PICP, \eta, \mu)} \tag{7}$$

where $\sigma(\cdot)$ is the sigmoidal defined as follows,

$$\sigma(PICP, \eta, \mu) = \frac{1}{1 + e^{\eta(PICP - \mu)}} \tag{8}$$

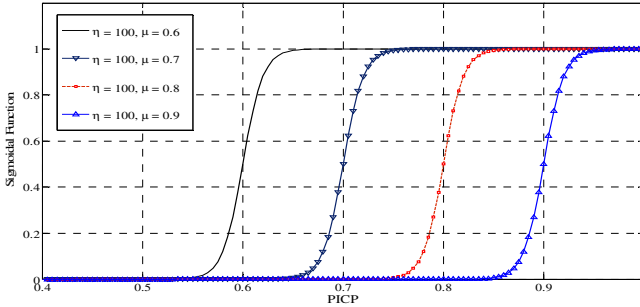


Fig. 1. The sigmoidal function for different values of η and μ

η and μ are two hyperparameters determined by modeler. Theoretically, PICP should be as close as possible to its nominal value, $(1 - \alpha)\%$, the confidence level that PIs have been constructed based on. This level of confidence can be appropriately used as a guide for selecting hyperparameters of CLC. One reasonable principle is that we highly penalize PIs that their PICP is less than $(1 - \alpha)\%$. This is based on the theoretical concept of PIs that their coverage probability in an infinite number of replicates will approach towards $(1 - \alpha)\%$.

Fig. 1 demonstrates $\sigma(\cdot)$ for different values of η and μ . It can be seen that the sigmoidal function sharply drops immediately after some values of PICP. These values are determined based on the confidence level of PIs, $(1 - \alpha)\%$. According to curves in Fig. 1, if PICP is less than some nominal thresholds, CLC will highly increase, no matter what the length of PIs is. In this way, PIs with not satisfactorily high coverage probability are highly penalized.

4 Optimization of PIs

In literature, PIs are constructed for NNs that have been trained based on minimization of an error-based cost function [2] [3]. In contrast to that research, the proposed method here on minimization of some measures/cost functions that are directly related to PI lengths and coverage probability. The cost function used in this research for optimization purposes is CLC defined in (7). In mathematical terms, the formulation is given as $\min_w \text{CLC}$, where w are NN parameters. To avoid problems related to over-fitting, data samples are split in two training sets. The first set is used for training NN in order to minimize an error-based cost function. The current NN parameters are then considered as the initial set in optimization process.

There are several options for minimization of CLC and, consequently, optimization of PIs. Mathematical analysis and minimization of CLC is quite difficult, mainly due to presence of very complex derivatives of NN output with respect to its parameters. Even if mathematical analysis of (7) is carried out, it is highly likely that the globally optimal solution is remote. This is mainly due to the fact that traditional techniques for training NNs, including backpropagation, are vulnerable to being trapped in local optimums of the multimodal search space. Those local optimums are inevitably present in many practical optimization problems, including NN parameter adjustment. Instead of these techniques, stochastic optimization methods can be employed for readjusting NN parameters based on (7). SA is a very powerful candidate for finding

the optimal values of NN parameters. Its stochastic nature allows it to explore different corners of the search space and escape the local optimums. Its application (with sufficient iterations) guarantees that NN parameters will move towards global optimality (or at least pareto optimal solutions) without being trapped in local optima. Optimization technique continues until one of the stopping criteria is met. Maximum number of iterations of the optimization, low speed of convergence, or satisfactory smallness of CLC are some stopping criteria used in our experiments. The procedure explained above can be summarized as follows:

- Step 1:* split data into two training sets,
- Step 2:* train the NN model in order to minimize a prediction error-based function,
- Step 3:* use the current NN parameters as the initial set and employ SA for minimizing CLC and readjusting NN parameters,
- Step 4:* examine performance of the trained NN for test samples.

Step 3 (which includes some sub-steps not reported here due to lack of space) is the key step in finding the optimal parameters of NNs. As the search space for NN parameters is multimodal, completion of this step is time-consuming. Computation mass is not a big deal in our study, as all steps can be carried out off-line.

Because the proposed method readjusts NN parameters based on minimization of CLC for a new set (2nd training set), not the one used for training NN for minimization of error-based cost functions (1st training set), it systematically includes avoidance of NN over-fitting. In this method, from one side, it is necessary to keep MSE for training set small as much as possible. From the other side, NN parameters are readjusted based on minimization of CLC evaluated for the 2nd training set. Because the optimization technique takes care of both MSE and CLC for two different sets, it is less likely that final NN will be over-fitted. Besides, as MSE is an important component used in the delta technique, it is always guaranteed that the modified NN yields a lower MSE for training set than does the original NN. This means that assumptions made in delta technique will remain all valid [3] [2]. Finally, because the proposed technique considers parameters obtained using minimization of the error-based cost function as its initial set, its superiority over traditional delta technique is guaranteed.

5 Simulation Results and Discussion

In this section, the proposed method is implemented for a real world case study described in [14] [15]. The underlying system is a medium-sized Baggage Handling System (BHS) with several autonomous and non-autonomous components, which are highly linked. The target in this study is time required for processing 90% of each flight bags. Database includes 272 samples which are divided in three sets: first training set (50%), second training set (25%), and test set (25%). Our experiments show that NN models are not appropriate for point prediction for these samples, due to presence of uncertainty in operation of BHS (probabilistic events including a bag being cleared in different levels of security check). Therefore PIs are constructed for these targets. A two layer NN with 7 neurons in the first layer and 4 neurons in second layer (72 parameters) is trained using the Levenberg–Marquardt algorithm. After completion of training (step 2), SA is implemented with the following

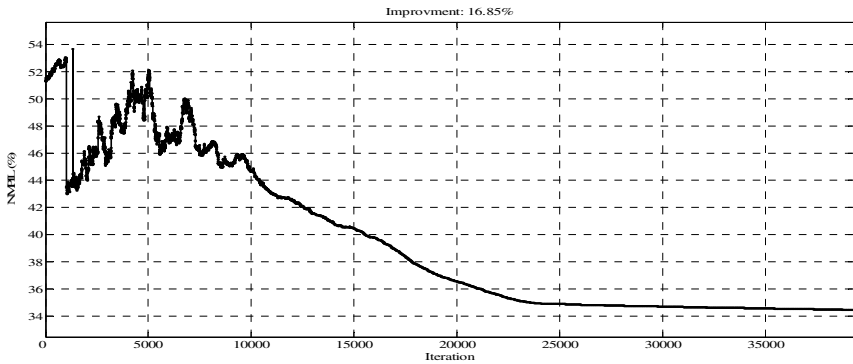


Fig. 2. CLC evolution during optimization process

parameters: the initial temperature is set to 10; a geometric cooling schedule is implemented by a cooling factor of 0.95; and in each temperature 100 rejection and success are allowed. η and μ in (8) are set to 100 and 0.875 respectively.

Fig. 2 represents how CLC varies during the optimization procedure (step 3) for the second training set. Optimization stops after around 39000 iterations. In the early iterations, because temperature is high (around 10), SA acts like a random search to enable exploration. Therefore, increase in CLC is welcomed. As temperature slowly decreases (through more iterations), SA behaviour becomes similar to a greedy hill-climbing algorithm (after iteration 7000) and converges to the optimal solution. Demonstrated results in Fig. 2 indicate that optimization technique makes it possible to reduce the cost function by more than 16.85%, without any loss in PICP. In fact, PICPs for 2nd training set computed using original NN and the optimized one are 91.18% and 88.23%. This growth again strongly confirms optimality of the set of new parameters (72 weights of NN model).

Performance of the proposed technique is also examined for unseen observations (step 4). PIs are constructed for NN retrained in step 3 using test samples. For the purpose of comparison, we also develop PIs using NN trained in step 2. Obtained results have been summarized in Table 1. These results show that NMPIL computed for the retrained NN is 7.19% less than NMPIL computed for the NN trained in step 2. This is an indication that performance of the delta technique can be improved through minimization of CLC rather than minimization of traditional error-based cost functions. Slight reduction in PICP from its nominal value is mainly due to dissimilarities of training and test sets. Also training set MSE for the original and optimized NN are 7.01 and 7.04, respectively. As these two quantities are quite close, assumptions made when developing theories of the delta technique have been remained valid in our experiments.

Table 1. Summary of results for test samples

	NN obtained in Step 2	NN obtained in Step 3
NMPIL	41.80	34.61
PICP	90.24	85.37
MSE (training samples)	7.01	7.04

6 Conclusion

In this paper, a novel criterion based on length and coverage probability of prediction intervals was developed for quantitative assessment of prediction intervals. This new criterion was appropriately applied for retraining neural network models that were used for constructing prediction intervals. Simulated annealing was integrated into the delta technique for readjusting neural network parameters in order to minimize the proposed measure. Demonstrated results for a real world case study showed that the proposed optimization method greatly improves quality of constructed prediction intervals not only for training samples, but also for future observations.

References

- [1] Dybowski, R., Roberts, S.J.: Confidence intervals and prediction intervals for feed-forward neural networks. In: *Clinical Applications of Artificial Neural Networks*, Cambridge, MA (2000)
- [2] Hwang, J.T.G., Ding, A.A.: Prediction Intervals for Artificial Neural Networks. *Journal of the American Statistical Association* 92, 748–757 (1997)
- [3] Veaux, R.D.d., Schumi, J., Jason, S., Ungar, L.H.: Prediction Intervals for Neural Networks via Nonlinear Regression. *Technometrics* 40, 273–282 (1998)
- [4] Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford (1995)
- [5] Lu, T., Viljanen, M.: Prediction of indoor temperature and relative humidity using neural network models: model comparison. *Neural Computing & Applications* 18, 345–357 (2009)
- [6] Yu, G., Qiu, H., Djurdjanovic, D., Lee, J.: Feature signature prediction of a boring process using neural network modeling with confidence bounds. *The International Journal of Advanced Manufacturing Technology* 30, 614–621 (2006)
- [7] Papadopoulos, G., Edwards, P.J., Murray, A.F.: Confidence estimation methods for neural networks: a practical comparison. *IEEE Transactions on Neural Networks* 12, 1278–1287 (2001)
- [8] Ho, S.L., Xie, M., Tang, L.C., Xu, K., Goh, T.N.: Neural network modeling with confidence bounds: a case study on the solder paste deposition process. *IEEE Transactions on Electronics Packaging Manufacturing* 24, 323–332 (2001)
- [9] Alonso, A.M., Sipols, A.E.: A time series bootstrap procedure for interpolation intervals. *Computational Statistics & Data Analysis* 52, 1792–1805 (2008)
- [10] Yang, L., Kavli, T., Carlin, M., Clausen, S., de Groot, P.F.M.: An evaluation of confidence bound estimation methods for neural networks. In: *Proceeding of ESIT* (2000)
- [11] Goffe, W.L., Ferrier, G.D., Rogers, J.: Global optimization of statistical functions with simulated annealing. *Journal of Econometrics* 60, 65–99
- [12] Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by Simulated Annealing. *Science* 220, 671–680 (1983)
- [13] Aarts, E., Korst, J.: *Simulated Annealing and Boltzmann Machine: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. J. Wiley, New York (1990)
- [14] Khosravi, A., Nahavandi, S., Creighton, D.: Estimating performance indexes of a baggage handling system using metamodels. In: *IEEE International Conference on Industrial Technology, ICIT 2009* (2009)
- [15] Khosravi, A., Nahavandi, S., Creighton, D.: Constructing Prediction Intervals for Neural Network Metamodels of Complex Systems. In: *International Joint Conference on Neural Networks, IJCNN 2009* (2009)

Robust Local Tangent Space Alignment

Yubin Zhan and Jianping Yin

School of Computer, National University of Defense Technology, Changsha, China
zhanyubin_dm@yahoo.com.cn

Abstract. This paper investigates noise manifold learning problem, which is a key issue in applying manifold learning to practical problem. A robust version of LTSA called RL TSA is proposed. The proposed RL TSA algorithm makes LTSA more robust from three aspects: firstly robust PCA algorithm is used instead of the standard SVD to reduce influence of noise on local tangent space coordinates; secondly RL TSA chooses neighborhoods that are approximated well by the local tangent space coordinates to align with the global coordinates; thirdly in the alignment step, the influence of noise on embedding result is further reduced by endowing clean data points and noise data points with different weights into local alignment errors. Experiments on both synthetic data sets and real data sets demonstrate the effectiveness of our RL TSA when dealing with noise manifold.

Keywords: manifold learning, robust PCA, local tangent space alignment.

1 Introduction

Local Tangent Space Alignment(LTSA) is an effective nonlinear dimensional reduction method proposed by Z. Zhang [3]. It shares the basic assumption of manifold learning [1,2,4,5] that high-dimensional data lie on a low-dimensional manifold. It can obtain expected result when this basic assumption for sampled data is true. Unfortunately, like other classical manifold learning algorithms, its sensitiveness to noise embarrasses it in real world applications. We illustrate its sensitiveness to noise by the following example.

Example 1. Considering 1500 examples sampled from the 2-D Swiss Roll manifold, we select 150 examples at random, and impose uniformly distributed noise to them. The fig. 1(a) plots these data points, where the black points are points with noise, and the colored point are clean data points. As can be seen from the fig. 1(b), due to noise, LTSA algorithm can't recover the manifold structure well.

Since the importance of robust manifold learning in real world application, some manifold learning algorithms have been extended to deal with the noise [6,7,8,9,10]. Most of these existing extensions work as an additional preprocess to detect noise and to reduce their influence before embedding algorithm performs. To our best

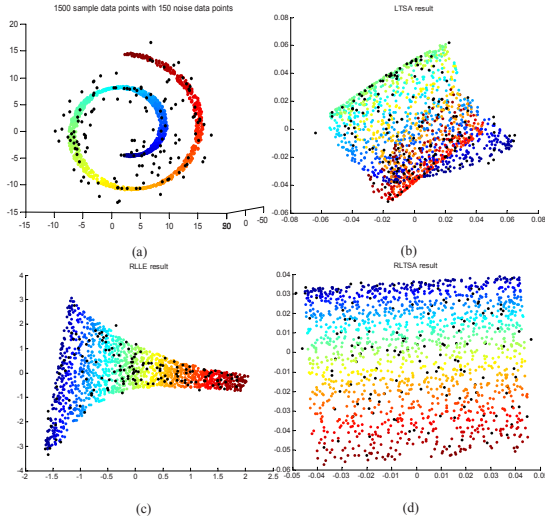


Fig. 1. Results of three algorithms on noise corrupted data set sampled from Swill Roll manifold. (a) data sets; (b) LTSA result; (c) RLLE result; (d) RL TSA result.

knowledge, there has no extension of LTSA to address the noise problem. Although MLTSA [11] proposed by J. Wang has the ability of dealing with noise on some local patches, since it mainly addresses LTSA’s failure modes on large curvature manifold, it has no ability to deal with noise distributed on whole manifold. This paper investigates noise manifold learning problem in context of LTSA and proposes a robust version of LTSA algorithm called RL TSA based on robust PCA. The main difference of our RL TSA with the previous robust manifold algorithm is that our denoise process is integrated seamlessly into the embedding algorithm and becomes part of it.

2 Reviews on LTSA

The basic idea of LTSA is that neighborhood of each point can be approximately represented by local tangent space coordinates. Given a data set $X = [x_1, x_2, \dots, x_n]$ sampled from a d -dimensional manifold \mathcal{M} where $x_i \in R^D (D > d)$ is column vector represented a sample and n is the number of samples. The basic steps of LTSA are as follows:

1. (Extracting local information) For each $x_i (i = 1, 2, \dots, n)$
 - (1) Constructing neighborhood $X_i = [x_{i_1}, x_{i_2}, \dots, x_{i_k}]$ (here we use k nearest neighbors including itself in terms of Euclidian distance in input space).
 - (2) By SVD of centered matrix $X_i - \bar{x}_i e^T$ where $\bar{x}_i = (1/k) \sum_{j=1}^k x_{i_j}$ and e is a column vector of all 1’s with suitable dimensionality, the orthogonal basis V_i of d -dimensional tangent space consists of the d left singular

vectors of matrix $X_i - \bar{x}_i e^T$ corresponding to the first d largest singular values. Then the local tangent space coordinates of neighborhood X_i can be computed as $\hat{X}_i = [\hat{x}_{i_1}, \hat{x}_{i_2}, \dots, \hat{x}_{i_k}] = V_i^T (X_i - \bar{x}_i e^T)$.

- (Constructing local alignment matrix) Denote the d -dimensional embedding coordinates by $Y = [y_1, y_2, \dots, y_n]$, and $Y_i = [y_{i_1}, y_{i_2}, \dots, y_{i_k}]$ which consists of the subset of column of Y is the corresponded d -dimensional embedding coordinates of neighborhood X_i . The local alignment matrix can be computed by minimizing the following local alignment error:

$$\min E_i = \min_{\substack{c_i \in \mathbb{R}^d \\ L_i \in \mathbb{R}^{d \times d}}} \sum_{j=1}^k \|y_{i_j} - (c_i + L_i \hat{x}_{i_j})\|^2 \tag{1}$$

$$= \min_{\substack{c_i \in \mathbb{R}^d \\ L_i \in \mathbb{R}^{d \times d}}} \|Y_i - (c_i e^T + L_i \hat{X}_i)\|_F^2 = \|Y_i \Phi_i\|_F^2 = \|Y S_i \Phi_i\|_F^2 \tag{2}$$

where S_i is the 0-1 select matrix such that $Y_i = Y S_i$ and $\|\cdot\|_F$ is the matrix Frobenius norm, we call Φ_i the local alignment matrix.

- (Aligning global coordinates) Computing the embedding coordinates Y by minimizing the following sum of local alignment error:

$$\sum_{i=1}^n \min E_i = \sum_{i=1}^n \|Y S_i \Phi_i\|_F^2 = \sum_{i=1}^n \text{tr}(Y S_i \Phi_i (Y S_i \Phi_i)^T) = \text{tr}(Y \Phi Y^T) \tag{3}$$

where $\Phi = \sum_{i=1}^n S_i \Phi_i \Phi_i^T S_i^T$ is the global alignment matrix. The solution is given by the eigenvectors of Φ .

From the basic steps of LTSA, one can see that recovering the real local tangent space is the key issue that relates to whether LTSA can discover the true manifold structure faithfully. In the presence of noise, however, the recovered tangent space by standard SVD technique will deviate from the real one due to its sensitivity to noise, then it will further influence embedding result of LTSA.

Note that the left singular vector of $X_i - \bar{x}_i e^T$ corresponding to the j -th largest singular value is the very eigenvector $v_j^{(i)}$ of covariance matrix $(X_i - \bar{x}_i e^T)(X_i - \bar{x}_i e^T)^T$ corresponding to the j -th largest eigenvalue, then the orthogonal basis of tangent space can be represented as $V_i = [v_1^{(i)}, v_2^{(i)}, \dots, v_d^{(i)}]$, so one can conclude that LTSA algorithm in fact uses the leading d -dimensional principal subspace to approximate the tangent space at each neighborhood. Therefore, robust PCA algorithm, which can find the robust principal subspace in presence of noise, can reduce the influence of noise on recovering the local tangent space.

When computing local alignment error, LTSA gives each point the same weight in equation (1). Obviously, in noise case, we should make distinction between clean points and noise points, and this can be implemented by specifying different weights to them.

In addition, there is no need to minimize the local align error sum of all neighborhoods [12][13][14] in equation (3). On the one hand, in no noise case

local tangent space coordinates derived from each neighborhood can character local geometric well, they are heavily redundant. On the other hand, in noise case, forcedly aligning the local coordinates of neighborhoods that are not well approximated will result in fatal error in the final embedding. Therefore, to make LTSA robust, one should discard the neighborhoods whose local coordinates can't character the local geometric well due to dominant effect of noise, and select neighborhoods that are approximated well by the local tangent space coordinates, then minimize the alignment error sum of these selected neighborhoods in equation (3).

According to the above analysis, our RL TSA algorithm will make LTSA more robust from three aspects: 1. robust PCA algorithm is used instead of the standard SVD to reduce effect of noise on local tangent space coordinates; 2. RL TSA selects neighborhoods that are well approximated by the local tangent space coordinates to align with the global coordinates, 3. in the alignment step, the influence of noise on embedding result is further reduced by specifying different weights to clean data points and noise data points in local align error.

3 The RL TSA Algorithm

Our RL TSA algorithm works as follows:

step 1: Construct neighborhood of each data point as the original LTSA algorithm does.

step 2:

1. Perform robust PCA on each neighborhood to obtain local tangent coordinates \hat{X}_i ;
2. determine weight w_i for each point in local alignment error formula (II);
3. select a neighborhood subset RN ;

step 3: For each neighborhood X_i in RN , obtain its local alignment matrix Φ_i via minimizing the following weighted local alignment error instead of equation (II):

$$\min_{\substack{c_i \in R^d \\ \hat{L}_i \in R^{d \times d}}} E_i = \min_{\substack{c_i \in R^d \\ \hat{L}_i \in R^{d \times d}}} \sum_{x_j \in X_i} w_j \|y_j - (c_i + \hat{L}_i \hat{x}_j^{(i)})\|^2 \tag{4}$$

It can be rewritten as the following matrix form:

$$\min_{\substack{c_i \in R^d \\ \hat{L}_i \in R^{d \times d}}} E_i = \min_{\substack{c_i \in R^d \\ \hat{L}_i \in R^{d \times d}}} \left\| \left(Y S_i - (c_i e^T + \hat{L}_i \hat{X}_i) \right) W S_i \right\|_F^2 \tag{5}$$

where $W = \text{diag}(\sqrt{w_1}, \sqrt{w_2}, \dots, \sqrt{w_n})$. As the same way in the original LTSA algorithm, the minimal weighted local alignment error can be also expressed as following form:

$$\min E_i = \|Y \Phi_i\|_F^2 \tag{6}$$

Then RL TSA obtain the global embedding coordinates via minimizing local alignment error sum of neighborhoods in RN :

$$\min \sum_{X_i \in RN} E_i = \min \sum_{X_i \in RN} \|Y\Phi_i\|_F^2 = \min \sum_{X_i \in RN} \text{tr}(Y\Phi_i\Phi_i^T Y^T) \quad (7)$$

The solution can be obtained by the eigendecomposition of the global alignment matrix $\Phi = \sum_{X_i \in RN} \Phi_i\Phi_i^T$.

In the following, we will give details of the key issues in our RL TSA algorithm: how to determine the weights and subset RN .

After performing robust PCA on each neighborhood, denote the local tangent coordinates, robust principal subspace and robust mean of neighborhood X_i by \hat{X}_i , V_i and μ_i respectively. Then for data point x_{ij} ($j = 1, 2, \dots, k$) in neighborhood X_i , its principal reconstruction error is:

$$\varepsilon_{ij} = \|x_{ij} - \mu_i - V_i \hat{x}_{ij}\| \quad (8)$$

Then a normalized error can be computed as $\varepsilon_{ij}^* = \varepsilon_{ij} / \sum_{j=1}^k \varepsilon_{ij}$. For each point x_j , we then compute its mean normalized error α_j over the neighborhoods that it is in. Set $\mathcal{N}_j = \{i : x_j \in X_i\}$, then $\alpha_j = 1/(\#\mathcal{N}_j) \sum_{i \in \mathcal{N}_j} \varepsilon_{ij}^*$ where ε_{ij}^* is x_j 's normalized error in neighborhood X_i . For each point x_j , its mean normalized error α_j can be serve as its likelihood as noise point. Set $\bar{\alpha} = \sum_{i=1}^n \alpha_i$ as the mean value of mean normalized errors. Then our RL TSA algorithm exploits the following formula to compute the weight of point x_j :

$$w_j = \begin{cases} 1, & \alpha_j \leq \bar{\alpha}; \\ \bar{\alpha}/\alpha_j, & \text{else.} \end{cases} \quad (9)$$

A natural request for selected neighborhoods is that they should contain more clean data points. Therefore we should select neighborhoods that contains relative more clean points. If we serve the points whose $\alpha_i \leq \bar{\alpha}$ as clean points, the percentage of clean points in neighborhood X_i is $\beta_i = \frac{\#\text{clean points in } X_i}{\#X_i}$.

Then the selected neighborhoods set \mathbb{R} can be expressed as:

$$RN = \{X_i | \beta_i \geq 0.6\} \quad (10)$$

4 Experimental Results

Extensive experiments on both artificial data sets and real world data sets have demonstrated the effectiveness of our RL TSA algorithm. In our implementation

¹ Although a few data points may be not covered by the selected neighborhoods, these points depart too far from the manifold, so that obtaining their embedding coordinates is meaningless, therefore removing it from the data set will not influence performance of our algorithm. Hereafter for brevity, we always assume all data points can be covered by the selected neighborhoods.

of RL TSA, we adopted the ROBPCA algorithm [15], which is an effective robust PCA technique and is extremely suitable for high-dimensional data, to perform robust PCA. Of course, other robust PCA method is also suitable.

4.1 Synthetic Data

We apply our RL TSA to data sets sampled from Swiss Roll manifold. To show the effectiveness of our RL TSA algorithm on noise manifold. we compare three algorithms LTSA, RLLE and our RL TSA. The results are shown in fig. 1. For data set sampled from Swiss Roll manifold which are embedding from R^3 to R^2 , the performance of algorithms can be seen by taking into account the coloring of the data points in the plots. From fig. 1, we can see that only the embedding results obtained by RLLE and our RL TSA vary the color smoothly, this means that only RLLE and RL TSA algorithm can recover the manifold structure well. So we can conclude that RL TSA algorithm can significantly improve the performance of the LTSA algorithm on noise corrupted data sets.

To evaluate the performance of RL TSA algorithm on different types of noise, we perform more experiments on data set sampled from Swill Roll manifold. This time we use Gaussian noise instead of the uniformly distributed noise. Meanwhile, we further study how the performance of RL TSA varies as the level of noise increases. Fig. 2 shows results of RLLE and RL TSA on data set with different level Gaussian noise respectively. We can see that when there has more

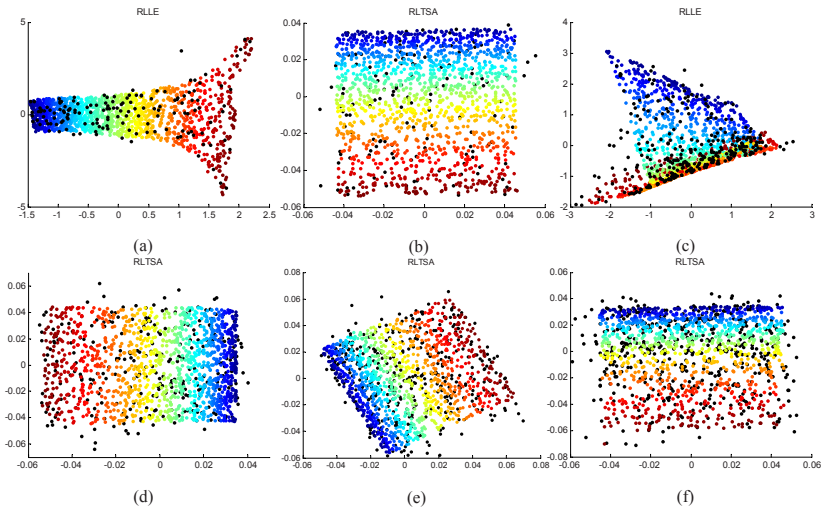


Fig. 2. Results of RLLE and RL TSA on Gaussian noise corrupted data sets sampled from Swiss Roll manifold. The data set contains 1500 points where we select different number of points at random and impose Gaussian noise with mean $\mu = 0$ and standard deviation $\sigma = 2$ on them. (a) RLLE result, 150 noise points; (b) RL TSA result, 150 noise points; (c) RLLE result, 300 noise points; (d) RL TSA result, 300 noise points; (e) RL TSA result, 450 noise points; (f) RL TSA result, 600 noise points.

than 20% noise points, RLLE is incapable of recovering the manifold structure. However, for our RL TSA, even if there has 40% noise points, it can still recover the manifold structure well. This demonstrates that our RL TSA algorithm outperform the state-of-the-art RLLE algorithm for noise corrupted data set.

4.2 Rendered Face Data Set

To illustrate the effectiveness of our RL TSA algorithm on high-dimensional real word data, we conduct experiments on rendered face data set² which is another benchmark data set used by many manifold learning algorithms. To generate noise images, we first randomly select 70($\approx 10\%$) images and for each selected image we change the value of randomly chosen 410 pixels($\approx 10\%$) by inverting each value(i.e., pixel value v is replaced by $1 - v$). Fig.3 shows ten original face images and their corresponding noise images. And the 2-D embedding result of RL TSA is shown in fig.4. One one see from it that the poses and light of embedding images(including noise images) vary smoothly, this means that our RL TSA algorithm preserves the intrinsic structure well.

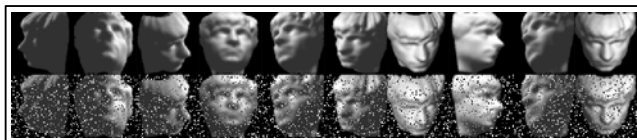


Fig. 3. Ten face images and their corresponding noise images

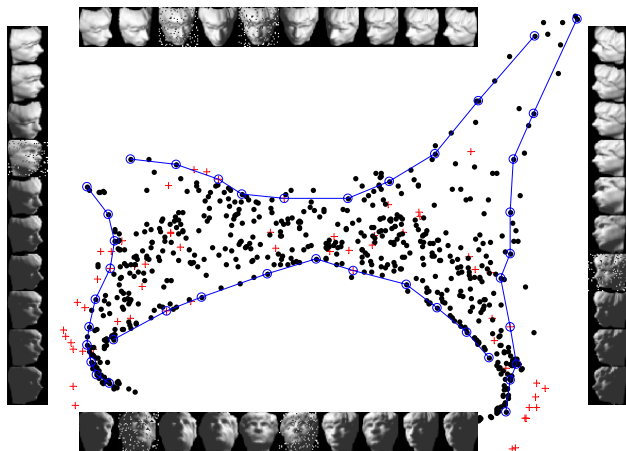


Fig. 4. 2-D embedding result of RL TSA algorithm. The “.” represents clean images, and red “+” represent noise images. Images correspond to the circled points linked by solid line.

² <http://isomap.stanford.edu>

To further quantitatively compare the performance of LTSA, RLLE and RL TSA, we use these algorithms to obtain 3-D embedding results of this face data set under different neighborhood size. Denote the matrix that consists of light parameters and poses parameters by P , and its centered matrix by $\hat{P} = P - (1/n)Pee^T$, then we can use the following relative reconstruction error to quantitatively evaluate performance of embedding algorithm:

$$error = \frac{\min_{L \in R^{3 \times 3}} \|\hat{P} - LY\|_F}{\|\hat{P}\|_F} \tag{11}$$

where Y is the embedding coordinates obtained by algorithm. Fig. 5 plots the relative reconstruction error of the 3-D coordinates computed by LTSA, RLLE and RL TSA under different neighborhood size respectively. One can clearly see that our RL TSA algorithm leads to significantly smaller relative reconstruction errors than others, this means our RL TSA can recover the pose and light parameter with higher accurately even in presence of noise. However, the relative errors of LTSA is close to 1, this means that in the presence of noise, LTSA can't recover the intrinsic parameters of the data set.

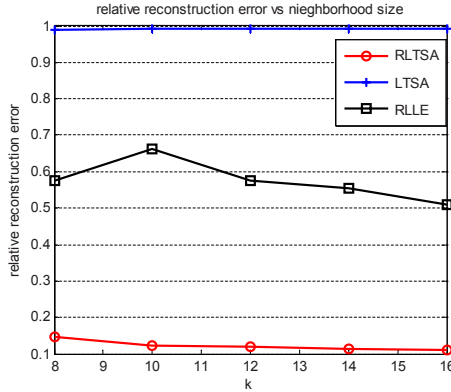


Fig. 5. Relative reconstruction errors of three algorithms under different neighborhood size

5 Conclusion

In this paper, a robust version of LTSA algorithm called RL TSA is proposed. Extensive experiments on artificial data sets and real world data set demonstrates the effectiveness of our RL TSA algorithm when dealing with noise corrupted data sets. Moreover the mechanism dealing with different types of neighborhoods and different data points can be used in other manifold learning algorithm to make them robust.

References

1. Tenenbaum, J.B., Silva, V.d., Langford, J.C.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* 290, 2319–2323 (2000)
2. Roweis, S.T., Saul, L.K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* 290, 2323–2326 (2000)
3. Zhang, Z., Zha, H.: Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment. *SIAM J. Scientific Computing* 26, 313–338 (2005)
4. Lin, T., Zha, H.: Riemannian Manifold Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 796–809 (2008)
5. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* 15, 1373–1396 (2003)
6. Chang, H., Yeung, D.-Y.: Robust locally linear embedding. *Pattern Recognit.* 39, 1053–1065 (2006)
7. Chen, H., Jiang, G., Yoshihira, K.: Robust nonlinear dimensionality reduction for manifold learning. In: *ICPR 2006*, NJ 08855-1331, United States, vol. 2, pp. 447–450. IEEE, Piscataway (2006)
8. Yin, J., Hu, D., Zhou, Z.: Noisy manifold learning using neighborhood smoothing embedding. *Pattern Recognit. Lett.* 29, 1613–1620 (2008)
9. Park, J., Zhang, Z., Zha, H., Kasturi, R.: Local smoothing for manifold learning, vol. 2, pp. 452–459. Institute of Electrical and Electronics Engineers Computer Society, Piscataway (2004)
10. Hein, M., Maier, M.: Manifold Denoising. In: *Advances in Neural Information Processing Systems*, vol. 19, pp. 561–568. MIT Press, Cambridge (2007)
11. Wang, J.: Improve local tangent space alignment using various dimensional local coordinates. *Neurocomputing* 71, 3575–3581 (2008)
12. Yang, L.: Alignment of Overlapping Locally Scaled Patches for Multidimensional Scaling and Dimensionality Reduction. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 438–450 (2008)
13. Zha, H., Zhang, Z.: Spectral analysis of alignment in manifold learning. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005)*, vol. 5, pp. 1069–1072 (2005)
14. Zha, H., Zhang, Z.: Spectral Properties of the Alignment Matrices in Manifold Learning. To appear in *SIAM Review* (2008)
15. Hubert, M., Rousseeuw, P.J., Vanden Branden, K.: ROBPCA: A new approach to robust principal component analysis

Probabilistic Combination of Multiple Evidence

Heeyoul Choi¹, Anup Katake², Seungjin Choi³,
Yoonseop Kang³, and Yoonsuck Choe¹

¹ Dept. of Computer Science and Engineering, Texas A&M University
3112 TAMU, College Station, TX 77843

{hchoi,choe}@cs.tamu.edu

² Starvision Technologies, Inc.

400 Harvey Mitchell Pkwy South, College Station, TX 77845

akatake@starvisiontech.com

³ Dept. of Computer Science, Pohang University of Science and Technology
San 31 Hyoja-dong, Nam-gu, Pohang 790-784, Korea

{seungjin,e0en}@postech.ac.kr

Abstract. In pattern recognition systems, data fusion is an important issue and evidence theory is one such method that has been successful. Many researchers have proposed different rules for evidence theory, and recently, a variety of averaging rules emerged that are better than others. In these methods, the key issue becomes how to give the weights to the multiple contributing factors, in order to calculate the average. To get better weights for the multiple bodies of evidence, we propose the use of structural information of the evidence. The bodies of evidence lie on a certain informational structure which can be described by a probability distribution and the probability of each evidence can serve as a weight for the evidence. Our experimental results show that our method outperforms other previous methods.

Keywords: Evidence Theory, Data Fusion, Decision Making, Probability, Belief Function.

1 Introduction

The nature and pace of advances in machine learning techniques is dramatically enhancing the effectiveness of pattern recognition methods. Many algorithms have been proposed for pattern recognition (see [1,2] and references therein). However, usually these algorithms are suitable to handle only one input signal source, even though the signal might be a mixture from multiple sources (i.e., a multivariate variable). When humans recognize some kind of pattern, they use multiple sensors and merge them together, or multiple persons might recognize something and then combine their opinions. This is because one sensor or a single person may not be good enough to unambiguously recognize something, and in this case more sensors or persons may lead to clearer and more stable recognition. Furthermore, the multiple sources (signals or humans) may have different levels of uncertainty associated with them. Therefore, in pattern recognition systems,

we need to handle such different levels of uncertainties from multiple sources or multiple recognition systems which could be implemented as neural networks (NNs) [3]. In this case, data fusion becomes an important issue, where Bayesian theory, fuzzy logic, and evidence theory are known to be effective, even though there is no consensus on which method is more universally applicable [4,5,6,7].

Evidence theory (ET) is a mathematically well defined theory for handling conflict between different bodies of evidence. It is conceptually the same as Bayesian theory except that it uses epistemic (subjective) uncertainty [8]. The advantages of ET include its flexibility in theory and easy implementability. In ET, a set of elements can be considered as a hypothesis with an associated degree of belief, and the sum of all beliefs does not have to be 1.0, unlike Bayesian methods where the sum of all probabilities should equal 1.0. After the initial introduction of ET by Dempster [5], it has been improved [6,9,7] because in some cases the original ET’s combination rule is against our intuitive reasoning. Many researchers have proposed different rules to address this issue, and recently, some effective averaging rules have emerged, and in these rules, how to assign the weights becomes an important issue [9,10,11]. These extensions of ET have been applied to many pattern recognition problems [12,13].

In this paper, we focus on an averaging method for the combination rule as proposed in [9,10,11]. We use the fact that multiple bodies of evidence give a probability distribution, and the probability of each piece of evidence on this distribution can serve as a weight for that evidence. Here, we simply use a Gaussian distribution as an approximation, to get the weights, and in turn calculate the average for the multiple bodies of evidence. We used the same data set from published averaging methods, and compared our method to those previous methods. Our experimental results are promising since our proposed method uses more information than other previous methods.

The rest of this paper is organized as follows. First, we briefly review ET and some averaging rules for ET in section 2. Then, in section 3, we propose a new probabilistic combination rule with discussions about its merit against the previous methods and its potential application to neural network systems. Section 4 shows two experimental results with analysis. Finally, we conclude our work with a brief outlook in section 5.

2 Related Work

2.1 Dempster-Shafer Theory

Dempster [5] proposed evidence theory and Shafer [6] developed it which led to *Dempster-Shafer* theory (D-S theory). Here, we give a brief review of the D-S theory. For details, see [7] and references therein.

Let Θ be a set of hypotheses, and m be a basic belief assignment (BBA) which is a function from a subset of Θ to $[0, 1]$ with the following properties.

$$\begin{aligned}
 m(\phi) &= 0, \\
 \sum_{A \subseteq \Theta} m(A) &= 1.
 \end{aligned}
 \tag{1}$$

When two evidence bodies m_1 and m_2 are given, the Dempster’s combination rule for $\tilde{m}(A)$ is defined by

$$\tilde{m}(A) = \frac{\sum_{B \cap C = A} m_1(B)m_2(C)}{1 - K}, \tag{2}$$

where

$$K = \sum_{B \cap C = \phi} m_1(B)m_2(C). \tag{3}$$

Here, K indicates basic probability associated with conflict. This can be easily expanded to more than two evidence bodies.

As pointed out in [9], in some cases Dempster’s combination rule is against our intuitive reasoning. For example, when only one evidence has 0 belief but all others have 1 belief, still the combination is 0. To overcome this weakness, ET has been improved in some directions such as Yager’s modified Dempster’s rule, Inagaki’s unified combination rule, Zhang’s center combination rule, Dubois and Prade’s disjunctive consensus rule, mixing or averaging, convolutive X-averaging and so on [7]. Among all these approaches, the averaging approach is known to be better than others [7][11].

2.2 Averaging Rules

In [9], Murphy proposed an averaging rule to avoid the nonintuitive combination in D-S theory as shown in the previous section. When there are N evidence bodies, Murphy’s rule first calculates the average of each hypothesis for the evidence. After calculating the averages, it applies the D-S combination rule with the averages $N - 1$ times. That is, Eq. (2) is modified as follows.

$$\tilde{m}(A) = \frac{\sum_{B \cap C = A} \bar{m}(B)\bar{m}(C)}{1 - \bar{K}}, \tag{4}$$

where

$$\bar{K} = \sum_{B \cap C = \phi} \bar{m}(B)\bar{m}(C). \tag{5}$$

Here, $\bar{m}(B)$ and $\bar{m}(C)$ are the averages of evidence for B and C , respectively. Note that it started using a first order statistics which is the average of the evidence. Here, all bodies of evidence have the same importance with the same weight in calculating the average, which is not always the case.

As in human decision making, each evidence needs to be assigned with a different weight. If one evidence is in harmony with other evidence, then it can be considered with high importance. Likewise, if one evidence is in high conflict, it can be considered less important. So, instead of a simple averaging rule, some other researchers have tried a weighted sum of evidence bodies [11][10]. Although their methods can not be easily summarized in a few equations, generally speaking, they use distances between evidence bodies for different weights, which can be interpreted as a second statistics of the evidence. These methods

have better performance than Murphy’s simple averaging method. However, the distance-based weight methods do not use all the information of the structure where the evidence bodies lie on. Also, they are not plugged into probability theory seamlessly and they are complex to implement.

3 Probabilistic Combination Rule

In this paper, we propose a new probabilistic combination rule for ET. Basically, as in [9] and [11,10], we calculate a weighted sum of evidence bodies for the representative value for each hypothesis from all the evidence. However, it is natural to assume that evidence bodies make a structure as in other data sets (see manifold learning methods [14,15,16]) and this structure can be described by a probability distribution. Then, we can use the probability of evidence on the distribution for different weights. Here, we calculate a new weighted sum which uses probability of the evidence.

Let $m_j(A_i)$ be the j th evidence of i th hypothesis, where $i = 1, \dots, C$ and $j = 1, \dots, N$. μ_i and σ_i^2 are the mean and the variance of i th hypothesis, respectively. As in the maximum likelihood (ML) estimate, we use a biased variance instead of an unbiased one, since the mean is also estimated. Moreover, the biased one gives more informative result especially with the small number of data points, even though the unbiasedness is a very attractive property [17, chap 4]. We assume a Gaussian distribution for the evidence bodies of each hypothesis to get a weight w_{ij} for $m_j(A_i)$ as follows.

$$w_{ij} = \frac{1}{Z_i} \exp\left\{-\frac{(m_j(A_i) - \mu_i)^2}{\sigma_i^2}\right\}, \tag{6}$$

where Z_i is a normalization term so that $\sum_j w_{ij} = 1$. Note that we cannot use a multivariate Gaussian model which might be able to use correlations between hypotheses, because the number of evidence might be less than that of the hypotheses. Now, the weight for each evidence is given by

$$\tilde{w}_j = \frac{1}{N} \sum_i w_{ij}. \tag{7}$$

Then, the weighted sum of the evidence bodies for the hypothesis A_i is obtained by

$$\tilde{m}_p(A_i) = \sum_j \tilde{w}_j m_j(A_i). \tag{8}$$

After calculating the weighted sums for all the hypotheses, we apply D-S combination rule $N - 1$ times as other averaging methods do. With Eq. (8), Eq. (2) is modified as follows.

$$\tilde{m}_p(A) = \frac{\sum_{B \cap C = A} \tilde{m}_p(B) \tilde{m}_p(C)}{1 - \bar{K}_p}, \tag{9}$$

where

$$\bar{K}_p = \sum_{B \cap C = \phi} \bar{m}_p(B) \bar{m}_p(C). \quad (10)$$

So, if one evidence has low probability in the evidence distribution, a very low weight is assigned to that evidence according to the probability. Likewise, an evidence with a high probability has high importance. In such a way, we use the information of the structure where the evidence bodies lie on and this is mathematically well defined even though the distribution model we assume here is simple.

Our proposed method uses probability of evidence instead of just mean or distances. Although the probability is based on corresponding Mahalanobis distance between the evidence and the mean when we use a simple Gaussian model, it is simply calculated by the distribution. We can expand this approach to more complicated distributions with many other density estimation methods such as a mixture of Gaussian model rather than a simple Gaussian distribution. So, our method is conceptually different from others, and physically this probability is more meaningful than the normalized distance for weights [18,19]. Also, probability is better than distance in terms of performance, which is confirmed in the next section.

In addition to combining the results from multiple recognition systems, our technique can be used for data fusion to help develop a more efficient and robust neural network system. For example, given two sets of measurements, the number of input nodes have to be doubled, making the system more complex. However, we can use our technique to combine the measurements, thus reducing the input layer size. Furthermore, our method can help remove noise or outliers through the data fusion process. As a result, the neural network can converge faster (fewer input nodes) and be more robust (noise resistant).

4 Experiments

In order to show the useful behavior of our method, we carried out experiments with two different data sets used in the previous published methods: (a) the data set in [11] (Data A) and (b) the data set in [10] (Data B). We compared our proposed method to their methods proposed in each paper. We implemented D-S theory, Murphy's averaging method and Chen's averaging method in [11] but we simply used Yong's results from the paper [10], for comparison with our results. Actually both cases are for target recognition systems where there is one true target (for both cases, the hypothesis A is the true target) with multiple evidence.

4.1 Data A

The belief table used in [11] is in Table 1. There are 5 evidence bodies and 3 BBAs for 3 hypotheses. Note that the second BBA for the hypothesis A is zero which is seriously conflicted with other evidence and evidence bodies 3, 4 and 5

have the same belief values. Intuitively, the hypothesis A should have a dominant belief after the combination rule and the hypothesis B should go close to zero. Also the influence of the second evidence is expected to be decreased as evidence bodies are added.

Table 1. Evidence of Data A

Belief	m_1	m_2	m_3	m_4	m_5
$m(A)$	0.50	0	0.55	0.55	0.55
$m(B)$	0.20	0.90	0.10	0.10	0.10
$m(C)$	0.30	0.10	0.35	0.35	0.35

Table 2. Comparison of combinations for Data A

Methods	Belief	$m_{1,2}$	$m_{1,2,3}$	$m_{1,\dots,4}$	$m_{1,\dots,5}$
Chen's	$m(A)$	0.1543	0.6026	0.8276	0.9048
	$m(B)$	0.7469	0.2239	0.0355	0.0061
	$m(C)$	0.0988	0.1735	0.1369	0.0891
Prob. Weights	$m(A)$	0.1543	0.7194	0.8594	0.9107
	$m(B)$	0.7469	0.0945	0.0078	0.0010
	$m(C)$	0.0988	0.1861	0.1327	0.0884

Table 2 shows the combination results of two methods as evidence bodies are added. We can see both methods have the hypothesis A going over 0.9 and the hypothesis B converging to almost zero after 5 evidence bodies are combined, which accords with our intuition. However, our proposed method converges faster than Chen's method as well as Murphy's, which we can see more easily in Fig. 1. In this figure, we can see that Murphy's method converges in a linear way

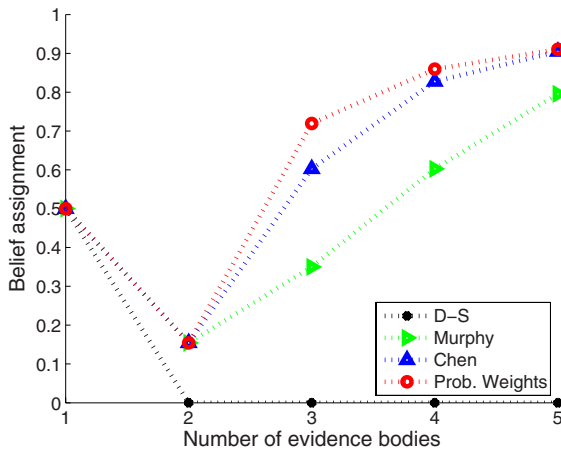
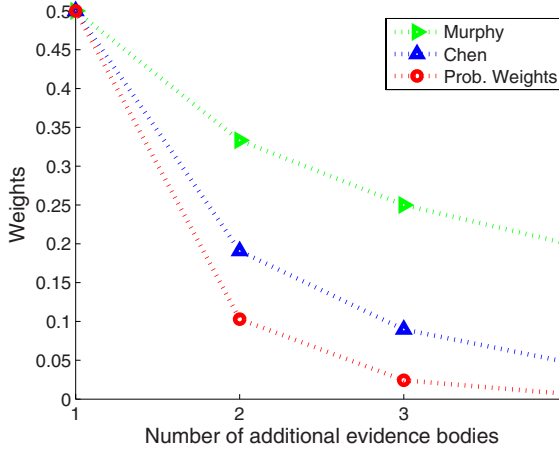
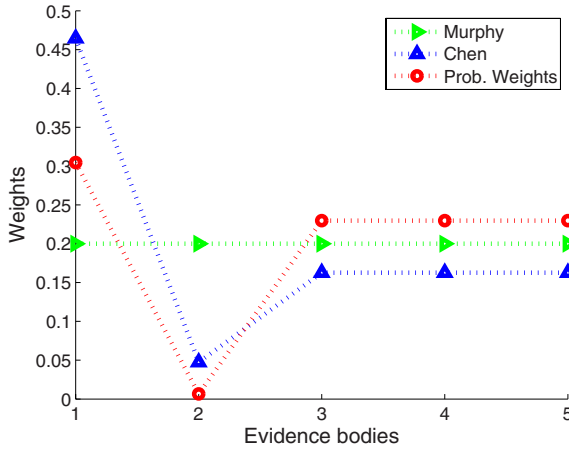


Fig. 1. The belief assignments of the hypothesis A from several methods for Data A



(a)



(b)

Fig. 2. The weights of three averaging methods: Murphy, Chen and Probabilistic Method. (a) The changing weight of the second evidence as the number of evidence increases, (b) The weights of all evidence bodies when 5 evidence bodies are given.

because it uses uniformly distributed weights, while Chen’s and our proposed method converge much faster than Murphy’s because they use the structure of evidence based on distances and probabilities, respectively. Note that the results of D-S combination for the hypothesis A are zero after evidence 2 no matter how high other BBAs are because it ignores all the conflicting evidence which can be interpreted as an AND operation as mentioned in [7].

Fig. 2 shows the weights for the evidence in three methods: Murphy’s, Chen’s and our proposed method. Fig. 2(a) shows how the weight for the second evidence

changes as other evidence bodies are added. The second evidence is seriously conflicted with others, so we want to minimize the effect (or weight) for it. Our proposed method depress the weight much faster so the effect of the second evidence gets more minimized than in other methods. Fig. 2(b) shows the weight of all evidence bodies after all evidence bodies are combined in three methods. In our proposed method, evidence 2 has less weight and evidence 3,4 and 5 have higher weights than other methods, which means our method finds out the proper weights aligned with our intuition.

4.2 Data B

The belief table used in 10 is in Table 3. As mentioned earlier, ET can have a set of elements as one hypothesis. In this table, there are 3 elements (A, B and C) and 4 hypotheses with ($\{A, C\}$) in addition to the 3 elements. There are 5 evidence bodies and the second one is seriously conflicted as in the previous data.

Table 3. Evidence of Data B

Belief	m_1	m_2	m_3	m_4	m_5
$m(A)$	0.5	0	0.55	0.55	0.6
$m(B)$	0.2	0.9	0.1	0.1	0.1
$m(C)$	0.3	0.1	0	0	0
$m(A, C)$	0	0	0.35	0.35	0.3

Table 4. Comparison of combinations for Data B

Methods	Belief	$m_{1,2}$	$m_{1,2,3}$	$m_{1,\dots,4}$	$m_{1,\dots,5}$
Yong's	$m(A)$	0.1543	0.4861	0.7773	0.8909
	$m(B)$	0.7469	0.3481	0.0628	0.0086
	$m(C)$	0.0988	0.1657	0.1600	0.1005
Prob. Weights	$m(A)$	0.1543	0.4768	0.9119	0.9879
	$m(B)$	0.7469	0.4518	0.0556	0.0031
	$m(C)$	0.0988	0.0656	0.0210	0.0039

In Table 4 and Fig. 3, our method converges faster than any other methods. Actually, Yong's method seems slightly better than ours and much better than Chen's when only 3 evidence bodies are combined, because the distribution is not well developed yet. However, from 4 evidence bodies, our method works much better than Yong's as well as any other methods. More interestingly, when 5 evidence bodies are combined, Yong's method is worst compared to all other methods, still our method is the best.

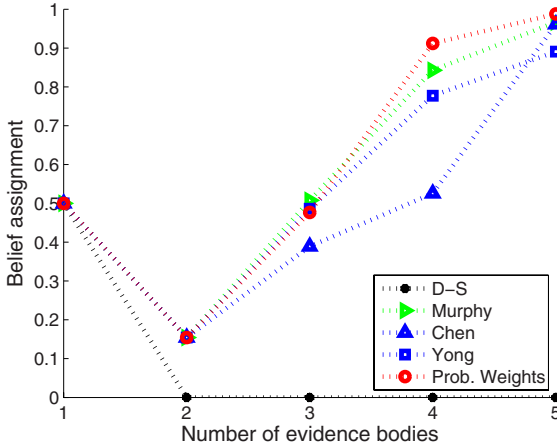


Fig. 3. The belief assignments of the hypothesis A from several methods for Data B

5 Conclusion

In this paper, we proposed a new way to calculate weights for the averaging method in evidence theory. Our proposed method uses the informational structure of evidence in the form of a probability distribution. Our method is well supported mathematically and conceptually, and is simple to implement. The performance of our method turned out to be superior to other existing methods.

A promising future direction is to replace the simplistic Gaussian distribution for the evidence to a more complex distribution. This will be especially necessary when the number of bodies of evidence is great.

Acknowledgments

Portion of this work was supported in part by KIPA under the program of Software Engineering Technologies Development and Experts Education, by CRC for Artificial Neurosensory Device and Cognitive System, and by KOSEF WCU Program (Project No. R31-2008-000-10100-0). Heeyoul Choi was supported by StarVision Technologies's student sponsorship program.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons, Chichester (2001)
2. Fukunaga, K.: An Introduction to Statistical Pattern Recognition. Academic Press, New York (1990)
3. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)

4. Hall, D.L., Llinas, J.: An introduction to multisensor data fusion. In: Proc. of the IEEE, vol. 85, pp. 369–376 (1997)
5. Dempster, A.P.: Upper and lower probabilities induced by a multivalued mapping. *The Annals of Statistics* 28, 325–339 (1967)
6. Shafer, G.: *SA Mathematical Theory of Evidence*. Princeton University Press, Princeton (1976)
7. Sentz, K., Ferson, S.: Combination of evidence in Dempster-Shafer theory. Technical Report Sandia report SAND2002-0835, Albuquerque, NM (2002)
8. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. *Machine Learning*, 131–163 (1997)
9. Murphy, C.K.: Combining belief functions when evidence conflicts. *Decision Support Systems* 29, 1–9 (2000)
10. Yong, D., WenKang, S., ZhenFu, Z., Qi, L.: Combining belief functions based on distance of evidence. *Decision Support Systems* 38, 489–493 (2004)
11. Chen, T., Que, P.: Target recognition based on modified combination rule. *Journal of Systems Engineering and Electronics* 17(2), 279–283 (2006)
12. Chen, A., Guan, Z.: Application of the information fusion based on evidence theory in urban environment. In: Proc. of SPIE, vol. 6043, pp. 131–137 (2005)
13. Xu, G., Tian, W., Qian, L., Zhang, X.: A novel conflict reassignment method based on grey relational analysis (GRA). *Pattern Recognition Letters* 28, 2080–2087 (2007)
14. Seung, H.S., Lee, D.D.: The manifold ways of perception. *Science* 290, 2268–2269 (2000)
15. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 2319–2323 (2000)
16. Roweis, S.T., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. *Science* 290, 2323–2326 (2000)
17. Hyvärinen, A., Karhunen, J., Oja, E.: *Independent Component Analysis*. John Wiley & Sons, Inc., Chichester (2001)
18. Choi, H., Choi, S., Choe, Y.: Manifold integration with Markov random walks. In: Proc. Association for the Advancement of Artificial Intelligence (AAAI), Chicago, IL (2008)
19. Kondor, R.I., Lafferty, J.: Diffusion kernels on graphs and other discrete structures. In: Proc. Int'l Conf. Machine Learning (2002)

FIA: Frequent Itemsets Mining Based on Approximate Counting in Data Streams*

Youngee Kim, Joonsuk Ryu, and Ungmo Kim

School of Information and Communication Engineering, Sungkyunkwan University,
300 Chunchun-dong, Suwon, Gyeonggi-Do, 440-746, Republic of Korea
youngees@gmail.com, scv82nim@gmail.com, umkim@ece.skku.ac.kr

Abstract. In this paper, we consider the problem of frequent elements over data stream seeks the set of items whose frequency exceeds σN for a given threshold parameter σ . We refer to this model as the sliding window model. We also use a user specified error parameter, ϵ , to control the accuracy of the mining result. We also propose an *FIA* (*Frequent Itemsets mining based on an Approximate counting*) algorithm based on the Chernoff bound with a guarantee of the output quality and also a bound on the memory usage. The proposed algorithm show that runs significantly faster and consumes less memory than do existing algorithms for mining approximate frequent itemsets.

Keywords: Frequent itemsets, Window Sliding, Chernoff bound, Approximate.

1 Introduction

In many applications, mining frequent itemsets on data streams is needed. It is thus of great interest to mine itemsets that are currently frequent. Several applications naturally generate data streams as opposed to data sets. In telecommunications, for example, call records are generated continuously. Typically, most processing is done by examining a call record once or operating on a “window” of recent call records, after which records are archived and not examined again. One of the challenging aspects of processing over data streams is that, while the length of a data stream may be unbounded, making it impractical or undesirable to store the entire contents of the stream, for many application [1]. Due to the constraints on both memory consumption and processing efficiency of stream processing, together with the exploratory nature of frequent itemsets mining, research studies have sought to approximate frequent itemsets over streams [2]. In the past few years, previous studies have been proposed to the efficient mining of frequent itemsets over data streams. The frequent elements problem over data stream seeks the set of items whose frequency exceeds σN for a given threshold parameter σ . Approximate mining algorithms use a related minimum support threshold (also called a user-specified error parameter), ϵ , where $0 \leq \epsilon < \sigma \leq 1$,

* This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government(MEST) (No. 2009-0075771).

to obtain an extra set of itemsets that are potential to become frequent later. A sliding window over a stream is a bag of last N elements of the stream seen so far, for some nonnegative integer N . This model captures recent pattern changes and trends. By the use of ε , we obtain highly accurate mining results and the mining efficiency is significantly improved. Existing approximation techniques for mining frequent itemsets are mainly false-positive approach [4-9]. Yu et al. proposed a false-negative approach [3]. The method focuses on the entire history of a data stream and does not distinguish recent itemsets from old ones. In this paper, we propose a false-negative approach in order to handle on recent data in a sliding window model. We also propose an *FIA* (*Frequent Itemsets mining based on Approximate counting*) algorithm based on the Chernoff bound with a guarantee of the output quality and also a bound on the memory usage. Therefore, our algorithm is controlled by two parameters ε and δ for error bounds and reliability. We can set a reasonable value for ε , accurate result, fast computation and low memory utilization can be achieved.

2 Problem Statement

In this section, we will prove a fairly general form of the Chernoff bound to mine frequent itemsets over data stream.

2.1 Preliminaries

Let X be a sum of n independent random variables $\{X_i\}$, with $E[X_i] = p_i$ such that $X_i \in \{0,1\}$ and $|X_i| \leq 1$ for all $i \leq n$. Let $X = \sum_{i=1}^n X_i$ and σ^2 be the variance of X and let μ denote the expected value of X . Then we have

$$\mu = E\left[\sum X_i\right] = \sum E[X_i] = \sum p_i \tag{1}$$

Then

$$\Pr\left[|X| \geq \lambda\sigma\right] \leq 2e^{-\lambda^2/4}, \quad \text{For any } 0 \leq \lambda \leq 2\sigma \tag{2}$$

Proof. By the above formula (2), we will prove as follows.

$$\Pr\left[|X| \geq \lambda\sigma\right] \leq 2e^{-\lambda^2/4} \tag{3}$$

The argument is symmetric for $\Pr[-X \geq \lambda\sigma]$. Let t be a real number between 0 and 1, to be determined later. Note that

$$\Pr\left[|X| \geq \lambda\sigma\right] = \Pr\left[tX \geq t\lambda\sigma\right] = \Pr\left[e^{tX} \geq e^{t\lambda\sigma}\right] \leq \frac{E[e^{tX}]}{e^{t\lambda\sigma}} \tag{4}$$

By the Markov inequality, we establish a bound on $E[e^{tZ}]$. Let $t \leq 1$ and $E[Z] = 0$, for all $-1 \leq Z \leq 1$. By the definition of expectation, since $|tz_k| \leq 1$, we can upper bound C .

$$\begin{aligned}
 \sum E[e^{tZ}] &= \sum_{k=1}^m p_k e^{tZ_k} \\
 &= \sum_{k=1}^m p_k \left(1 + tZ_k + \frac{1}{2!}(tZ_k)^2 + \frac{1}{3!}(tZ_k)^3 + \dots + \frac{1}{m!}(tZ_k)^m \right) \\
 &= \sum_{k=1}^m p_k + t \sum_{k=1}^m p_k Z_k + \sum_{k=1}^m p_k \left(\frac{1}{2!}(tZ_k)^2 + \frac{1}{3!}(tZ_k)^3 + \dots + \frac{1}{m!}(tZ_k)^m \right) \\
 &= \sum_{k=1}^m p_k (tZ_k)^2 \left(\frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{m!} \right) \leq t^2 \sum_{k=1}^m p_k Z_k^2
 \end{aligned} \tag{5}$$

Let $(1 + t^2 \text{Var}[X_i])$ be a sum of variance of Z . In equation (4), $t = \lambda / 2\sigma$

$$\begin{aligned}
 E[e^{tX}] &= E[e^{t(X_1 + X_2 + \dots + X_n)}] \\
 &= E[\prod_{i=1}^n e^{tX_i}] = \prod_{i=1}^n E[e^{tX_i}] \\
 &\leq \prod_{i=1}^n (1 + t^2 \text{Var}[X_i]) \leq \prod_{i=1}^n e^{t^2 \text{Var}[X_i]} = e^{t^2 \sigma^2}
 \end{aligned} \tag{6}$$

Thus, we get that the expected value of X .

$$\Pr[X \geq \lambda \sigma] \leq \frac{e^{t^2 \sigma^2}}{e^{t \lambda \sigma}} = e^{t \sigma (t \sigma - \lambda)} \leq e^{-\lambda^2 / 4} \tag{7}$$

2.2 Chernoff Bound

Suppose there is a sequence of elements, $e_1, e_2, \dots, e_i, \dots, e_N$, in data stream and consider the first n ($n \ll N$) observations as independent Bernoulli trails(coin flips) such that $Pr(head) = p$ and $Pr(tail) = 1 - p$ for a probability p . Let k be the number of heads in the n coin flips. Then, the expectation of k is np . Chernoff bound states, for any $\sigma > 0$. From equation (7),

$$\Pr[k - np \geq np\sigma] \leq e^{-\sigma^2 / 4} \leq 2e^{-np\sigma^2 / 2} \tag{8}$$

By substituting $\bar{k} = k/n$ and $\varepsilon = p\sigma$,

$$\begin{aligned}
 \Pr \left[|\bar{k} \cdot n - np| \geq n\varepsilon \right] &\leq 2e^{-np\sigma^2 / 2} \quad (\text{by } \sigma = \varepsilon/p) \\
 &= \Pr \left[|\bar{k} - p| \geq \varepsilon \right] \leq 2e^{-np\sigma^2 / 2} \leq 2e^{-\frac{n\varepsilon^2 / p}{2}} \leq 2e^{-n\varepsilon^2 / 2p}
 \end{aligned} \tag{9}$$

Let $\delta = 2e^{-n\varepsilon^2 / 2p}$. We obtain the following equation.

$$\log_{2, e^\delta} = \log_{2, e^{-n\varepsilon^2 / 2p}}, \quad \frac{-n\varepsilon^2}{2p} = \frac{\log_2 \delta}{\log_2 2e} = \log_{2, e} \delta, \quad -n\varepsilon^2 = 2p \cdot \log_{2, e} \delta, \quad \varepsilon^2 = \frac{2p \cdot \log_{2, e} \delta}{-n} \tag{10}$$

$$\varepsilon = \sqrt{\frac{2p \cdot \log_{2, e} \delta}{n}} = \sqrt{\frac{2p \cdot \ln(\frac{\delta}{2})}{n}}$$

By $s = p$, the minimum support s as the probability p .

$$\epsilon = \sqrt{\frac{2s \cdot \ln(\frac{2}{\delta})}{n}} \tag{11}$$

Then, from equation (9), we can produce \bar{k} satisfying:

$$Pr[s - \epsilon \leq \bar{k} \leq s + \epsilon] \geq 1 - \delta \tag{12}$$

In other word, for a itemset X , true support of X is within $\pm\epsilon$ of s with reliability $1 - \delta$. In order to test whether our false negative oriented approach decrease error propagation, setting minimum support and reliability in Table 1. We can see that, our bound does not rely on the user-specified σ , but on a chernoff bound ϵ which decreases while the number of observations n increases. As a result, we believe that our method, promising to solve our algorithm.

Table 1. Error propagation of false negative (ϵ) and false positive($\sigma=s/10$) with stream size(n) in the range $[s, \delta]$

$[s, \delta]$	Size (n)	ϵ	$\sigma=s/10$	$[s, \delta]$	Size (n)	ϵ	$\sigma=s/10$
[0.1, 0.1]	3518	0.0131	0.01	[0.2, 0.1]	3518	0.0185	0.02
[0.1, 0.1]	5991	0.0100	0.01	[0.2, 0.1]	5991	0.0141	0.02
[0.1, 0.1]	11278	0.0073	0.01	[0.2, 0.1]	11278	0.0103	0.02
[0.1, 0.1]	64085	0.0031	0.01	[0.2, 0.1]	64085	0.0043	0.02
[0.1, 0.1]	93263	0.0025	0.01	[0.2, 0.1]	93263	0.0036	0.02
[0.1, 0.1]	1182911	0.0007	0.01	[0.2, 0.1]	1182911	0.0010	0.02

3 Proposed Algorithm

In this section, we develop an algorithm based on the chernoff bound for mining frequent itemsets, called FIA. This algorithm offer to devise techniques for storing summary or synoptic information about previously seen portions of data stream. Hence the proposed method give a tradeoff between the count of some approximate frequent itemsets and the count of real frequent itemsets to provide precise answers to involve past data. We refer to this model as the sliding window model.

3.1 Basic Concept of FIA Method

Assume that the current data stream, $DS_m = \{e_1, e_2, \dots, e_m\}$ when the current size of the stream is m and the current size of window is N . In addition, when the size of a sliding window is denoted by W , the current window $DS_{(W,N)} = \{e_{m-N+1}, e_{m-N+2}, \dots, e_m\}$ in the current data stream DS_m is defined by the set of N transactions that are most recently generated. In our algorithm, we divide the itemsets into three groups – *frequent*

itemsets in a current window of size N , potential frequent itemsets and unpromising infrequent itemsets in a current window of size N . Basically, the process of FIA algorithm is shown as follows. In step 1, *window initialization phase*, is activated that after $m \geq N (= W)$ elements of the stream. In this phase, each element of the new incoming transaction is approximately counted. A $DS_{(w,N)}$ allow $\varepsilon \cdot e / N$ approximate counts to be computed over the current window. In step 2, *Frequent itemsets generation phase*, by the user support threshold σ ($0 \leq \sigma \leq 1$) and computed chernoff bound ε ($0 \leq \varepsilon < \sigma$) in a current window find frequent itemsets. For any a approximate frequent itemset \tilde{F}_i , (1) \tilde{F}_i is frequent if $a_count(\tilde{F}_i) \geq \sigma N$, (2) is potential frequent if $\varepsilon N \leq a_count(\tilde{F}_i) \leq \sigma N$, and (3) \tilde{F}_i is infrequent if $a_count(\tilde{F}_i) < \varepsilon N$. The potential frequent itemset may become frequent later. We are only interested in frequent itemsets, and infrequent itemsets will discard because the number of infrequent itemsets is really large over data stream. Hence, the error will be no more than ε because of the loss of support from infrequent itemsets. It is guarantees that no false negative. Finally, in step 3, *Updating window*, the arrival of a new block also triggers in current window, which are differently executed in three cases: (1) New itemset insertion (2) Old itemset update (3) Itemset discounting.

3.2 Mining of Potential Frequent Itemsets within a Current Sliding Window

The frequent itemsets generation is described as follows. First of all, our algorithm FIA read every transaction from the current block of current window. Then, we keep the potential frequent itemsets in the active table with respect to σ in each block B . For the first block B , after finding the potential frequent itemsets, it is set to the support of each itemsets. For example, in Fig 1, the first sliding window W_1 contains the three blocks: B_1, B_2, B_3 . Let the user-defined minimum support threshold σ be 0.5. First, FIA reads the first block B_1 and support counter is set in the active table. Then, the itemsets is frequent if $a_count(\tilde{F}_i) \geq \sigma N$ where itemset "a" is $6 \geq 0.5 * 10$. In initialization step, the p_count in potential table is set to be 0. If $a_count(\tilde{F}_i) < \sigma N$ then we keep frequency counts into potential table. After pruning all infrequent information from the active table, and its support counter contains into potential table. The proposed algorithm is performed using apriori property in order to find the frequent k -itemsets. The candidate generation process is stopped until no new candidates frequent itemsets with $k+1$ itemsets are generated. The result is shown in Fig 1.

Next, when the current sliding window W_1 is becomes full, merged to itemsets with support of longest frequent itemsets in each blocks as Fig 2.

Therefore, the generated frequent itemsets in current window W_1 becomes {a, b, c, ab, bc} as shown in Fig 3 and we know there are at most a equation (6) support count of frequent itemsets in the current window.

$$\sigma \cdot \sum_{i=1}^{|W_1|} B_i = \sum_{i=1}^{|W_1|} \max[a_count(B_i)] / |W_1| \quad (13)$$

BLOCK [B₁]

TID	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆	T ₇	T ₈	T ₉	T ₁₀
items	abcd	abcde	bc	bcde	abcde	bcd	ad	abcd	bde	abc

active table

itemset	a	b	c	d	e
a_count	6	9	8	8	4

potential table

itemset	a	b	c	d	e
p_count	0	0	0	0	0

frequent itemsets

itemsets	a	b	c	d	ab	ac	ad	bc	bd	cd	abc	bcd
a_count	6	9	8	8	5	5	4	6	7	6	5	6

active table

itemset	a	b	c	d
a_count	6	9	8	8

potential table

itemset	a	b	c	d	e	ad
p_count	0	0	0	0	4	4

(a) Frequent itemsets generation in block 1 after sliding

BLOCK [B₂]

TID	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆	T ₇	T ₈	T ₉	T ₁₀
items	bcd	abe	abc	bce	abce	bce	ade	abc	be	abcde

active table

itemset	a	b	c	d	e
a_count	6	9	7	7	7

potential table

itemset	a	b	c	d	e
p_count	0	0	0	0	3

frequent itemsets

itemsets	a	b	c	e	ab	bc	bc	bc	be	ce
a_count	6	9	7	7	5	4	4	7	6	4

active table

itemset	a	b	c	e
a_count	6	9	7	7

potential table

itemset	a	b	c	d	e	ad	ac	ae	ce
p_count	0	0	0	3	4	4	4	4	4

(b) Frequent itemsets generation in block 2 after sliding

BLOCK [B₃]

TID	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆	T ₇	T ₈	T ₉	T ₁₀
items	bc	ace	abc	abe	bce	ace	abd	abc	be	abce

active table

itemset	a	b	c	d	e
a_count	7	8	7	7	6

potential table

itemset	a	b	c	d	e
p_count	0	0	0	0	1

frequent itemsets

itemsets	a	b	c	e	ab	ac	bc	bc	bc	ce
a_count	7	8	7	6	5	5	4	5	4	4

active table

itemset	a	b	c	e
a_count	7	8	7	6

potential table

itemset	a	b	c	d	e	ad	ac	ae	ce	be
p_count	0	0	0	3+1=4	4	4	4	4	4+4=8	4+4=8

(c) Frequent itemsets generation in block 3 after sliding

Fig. 1. Steps of frequent itemsets generation to each sliding block in current window W_1

MERGE $B_1 + B_2 + B_3$

[B_1 : frequent itemsets]											
itemsets	a	b	c	d	ab	ac	bc	bd	cd	abc	bcd
a_count	6	9	8	8	5	5	6	7	6	5	6

[B_2 : frequent itemsets]									
itemsets	a	b	c	e	ab	bc	be		
a_count	6	9	7	7	5	7	6		

[B_3 : frequent itemsets]							
itemsets	a	b	c	e	ab	ac	bc
a_count	7	8	7	6	5	5	5

[Merge: $B_1 + B_2 + B_3$]	
a	$5 + 5 + \max[5,5] = 15/ W = 5$
b	$\max[5,6] + \max[5, 7, 6] + \max[5,5,5] = 18/ W = 6$
c	$\max[5,6] + 7 + \max[5,5] = 18/ W = 6$
d	$6 + 0 + 0 = 6/ W = 2$
e	$0 + 6 + 0 = 6/ W = 2$
ab	$5 + 5 + \max[5,5] = 15/ W = 5$
ae	$5 + 5 = 10/ W = 3.3$
bc	$\max[5,6] + 7 + \max[5,5] = 18/ W = 6$

Fig. 2. Steps of block merge in current window W_1

[Current window (W_1): frequent itemsets]					
itemsets	a	b	c	ab	bc
a_count	5	6	6	5	6

Fig. 3. Generated frequent itemsets in current window W_1

3.3 Mining Frequent Itemsets Insert and Delete Phase

The problem of mining frequent itemsets in recent data streams is to mine the set of all frequent itemsets by one scan. In our algorithm, after the oldest block is removed from the current sliding window, a new incoming block is appended to the window. As shown Fig 4, in insert states, each frequent itemset is accumulated approximate count and the potential count in the current block. Then, we estimate the maximum possible sum of its approximate support counts in the subsequent block based on the ϵ value. For each itemset that is in current window but not frequent in B_i , we compute its support count in B_i by scanning the buffer to update its *a_count*. In Fig 5, after an itemset in current window is deleted if its sum of *a_count* and *p_count* is less than $\epsilon \cdot \sum B_i$. Since, the transactions in $B_{t-|W|}$ will the expired, the support counts of the itemsets kept by current window are discounted accordingly.

3.4 Pruning Strategy

In the FIA algorithm, the constant value $\sigma \cdot |N|$ is the frequent threshold of itemsets, where σ is the user-defined minimum support threshold. It is important to note that ϵ

is not the user specified parameter but a running variable. The running error ϵ decreases, while the size of window W increases. Therefore, $\widetilde{F}_i \approx sN$.

[Insert: B₄]

TID	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆	T ₇	T ₈	T ₉	T ₁₀
items	abe	ace	abce	ade	bce	ace	abe	abe	bce	abce
[Current window (W₁): frequent itemsets]										
itemsets	a		b		c		ab		bc	
a_count	5		6		6		5		6	
<i>active table</i>										
itemset	a		b		c		d		e	
a_count	8 + 5 = 13		7 + 6 = 13		6 + 6 = 12		2		9 + 4 = 13	
[W₁+B₄: frequent itemsets]										
itemsets	a	b	c	e	ab	ac	ae	bc	bc	ce
a_count	13	13	12	13	10	4	7	10	6	6
<i>potential table</i>										
itemsets	a	b	c	d	e	ad	ac	ae	ce	be
p_count	0	0	0	4+2=6	40	4	4+4=8	4+4=8	4+4=8	4

Fig. 4. Steps of insert to new block in current window W_1

[Delete: B₁]

[W₁+B₄: frequent itemsets]											
itemsets	a	b	c	e	ab	ae	bc	bc	ce		
a_count	13	13	12	13	10	7	10	6	6		
[B₁: frequent itemsets]											
itemsets	a	b	c	d	ab	ac	bc	bd	cd	abc	bcd
a count	6	9	8	8	5	5	6	7	6	5	6
[W₁+B₄-B₁: frequent itemsets]											
<i>active table</i>											
itemset	a	b	c	d	e						
a count	13 - 5 = 8	13 - 6 = 7	12 - 6 = 6		2					13 - 0 = 13	
<i>[frequent itemsets]</i>											
itemsets	a	b	c	e	ab	ac	ae	bc	bc	ce	
a_count	8	7	6	13	7	6	8	6	7	6	
<i>[potential table]</i>											
itemsets	a	b	c	d	e	ad	ac	ae	ce	be	
p_count	0	0	0	6	0	4	0	0	0	0	

Fig. 5. Steps of delete to first b2lock in current window W_1

The pseudo code of algorithm FIA is outlined below. Our algorithm is conducted the impacts of a large number itemsets in the range of $[\sigma - \epsilon, \sigma + \epsilon]$ on frequent itemsets mining over active window. We obtain ϵ based on the chernoff bound and both

are initialized potential frequent itemset support count p_count , and use a_count for approximate count in active window. When B_i arrives, where $1 \leq i \leq |W|$, three conditions are executed one by one from 4 to 14.

Algorithm: FIA

Input: DS_{db} (a transaction data stream), σ (a user-defined minimum support threshold in the range of $[0,1]$), δ (a user-defined probability), $|W|$ (window size), ϵ (chernoff bound)

Output: A set of frequent itemsets \widetilde{F}_i

```

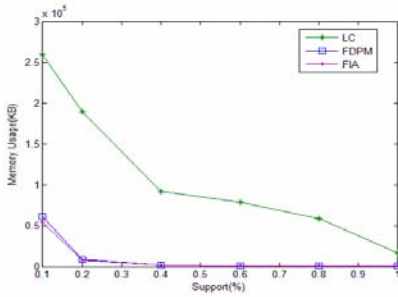
1. Begin
2.  $a\_count = 0; p\_count = 0; W = \text{NULL};$ 
3. for each new transaction  $T_i$  in  $W$  do
4.   if  $W = \text{FULL}$  then
5.     for all itemset  $e$  in  $T_i$  do
6.        $a\_count.cnt = a\_count.cnt + 1;$ 
7.       if  $(a\_count.cnt < \sigma)$ 
8.          $delete\ a\_count.table(e)$ 
9.          $p\_count.cnt = a\_count.cnt;$ 
10.    end for
11.    $\widetilde{F}_i(k=1) = \{\text{frequent 1-itemsets}\};$ 
12.   for  $(k=2; \widetilde{F}_i(k-1) \neq \text{NULL}; k++)$  do
13.     Generate the frequent  $k$ -itemsets;
14.   end for
15.   for each new block do
16.     /* A new incoming block */
17.     current window.a_count + new block.a_count
18.     if  $(a\_count.cnt < \sigma)$ 
19.       delete a_count.table( $e$ )
20.        $p\_count.cnt = a\_count.cnt;$ 
21.     end for
22.     delete oldest block. a_count, p_count;
23.   Output  $\widetilde{F}_i$  ( $a\_count \geq \epsilon N$ )

```

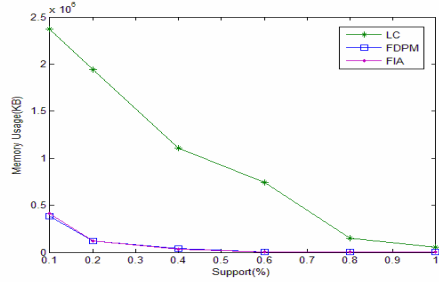
4 Experimental Results

In this section, we will describe the experimental evaluation of the proposed algorithms, FIA. We evaluate the performance of our FIA algorithm by varying the usage of the memory space. We also analyze the execution time. The simulation is implementation in Visual C++ and conducted in a machine with 3GHz CPU and 1GB memory. We use two sets of synthetic databases by using IBM Quest data generator. Two synthetic data streams, denoted by T10I4D1000K and T40I10D1000K are generated, where some of the parameters mean that the average size of the transaction T , the average size of the frequent itemsets I , and the total number of transaction D . In the following experiments, the minimum support threshold σ vary from 0.1% to 1.0%, $\delta = 0.05$ in data sets. The size of the sliding window is 20K transactions. We compare our algorithm FIA with Lossy Counting and FDPDM algorithm. As shown in Fig 6, FIA significantly outperforms LC. From the figures we can see that the memory requirement of proposed algorithm in the frequent itemset mining process is less than that of LC and FDPDM. Fig 7 shows the processing time on two data sets.

We can see that as the support increases, the processing time of frequent itemsets mining for all algorithms decreases. The processing time of our FIA algorithm is faster than that of LC and FDPDM. Therefore, the proposed our algorithm is a time and memory efficient method for frequent itemsets mining from data streams based on window sliding. Our experimental results support false negative approach.

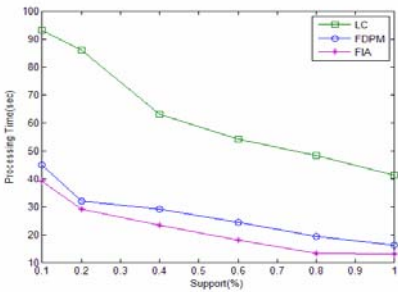


(a) T10I4D1000K

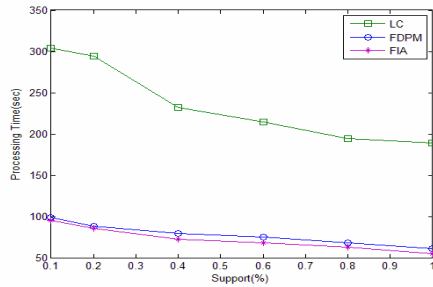


(b) T40I10D1000K

Fig. 6. Comparisons of memory usages in varying support σ



(a) T10I4D1000K



(b) T40I10D1000K

Fig. 7. Comparisons of processing time in varying support σ

5 Conclusion

In this paper, we study the problem a false-negative approach in order to handle on recent data in a sliding window model. We also propose an FIA algorithm based on the Chernoff bound with a guarantee of the output quality and also a bound on the memory usage. Therefore, our algorithm is controlled by two parameters ϵ and δ for error bounds and reliability. We can set a reasonable value for ϵ , accurate result, fast computation and low memory utilization can be achieved. We evaluate the performance of our FIA algorithm by varying the usage of the memory space. We can see that the memory requirement of proposed algorithm in the frequent itemset mining process is less than that of LC and FDPM.

References

1. Datar, M., Gionis, A., Indyk, P., Motwani, R.: Maintaining stream statistics over sliding windows. *SIAM Journal on Computing* 31(6), 1794–1813 (2002)
2. Manku, G.S., Motwani, R.: Approximate Frequency Counts Over Data Streams. In: *Proceedings of the 28th International Conference on VLDB*, pp. 346–357 (2002)

3. Yu, J.X., Chong, Z., Lu, H., Zhang, Z., Zhou, A.: False positive or false negative: mining frequent itemsets from high speed transactional data streams. In: Proc, VLDB (2004)
4. Chang, J., Lee, W.: A Sliding Window Method for Finding Recently Frequent Itemsets over Online Data Streams. *Journal of Information Science and Engineering* 20 (2004)
5. Lee, C.H., Lin, C.R., Chen, M.S.: Sliding window filtering: An efficient method for incremental mining on a time-variant database. *Information Systems* 30, 227–244 (2005)
6. Lin, C.-H., Chiu, D.-Y., Wu, Y.-H., Chen, A.L.P.: Mining frequent itemsets from data streams with a time-sensitive sliding window. In: Proc, SIAM Int'l Conference on Data Mining, pp. 68–79 (2005)
7. Giannella, C., Han, J., Pei, J., Yan, X., Yu, P.S.: Mining frequent patterns in data streams at multiple time granularities. In: *Data Mining, Next Generation Challenges and Futures Directions*, pp. 191–212. AAAI/MIT Press (2004)
8. Li, H.F., Lee, S.Y.: Mining frequent itemsets over data streams using efficient window sliding techniques. *Expert Systems with Applications* (2008)
9. Li, H.F., Ho, C.C., Shan, M.K., Lee, S.Y.: Efficient Maintenance and Mining of Frequent Itemsets over Online Data Streams with a Sliding Window. In: *IEEE SMC 2006* (2006)

Advances in PARAFAC Using Parallel Block Decomposition

Anh Huy Phan and Andrzej Cichocki*

Lab for Advanced Brain Signal Processing
Brain Science Institute - Riken
Wako-shi, Saitama 351-0198, Japan
{phan, cia}@brain.riken.jp

Abstract. Parallel factor analysis (PARAFAC) is a multi-way decomposition method which allows to find hidden factors from the raw tensor data with many potential applications in neuroscience, bioinformatics, chemometrics etc [1,2]. The Alternating Least Squares (ALS) algorithm can explain the raw tensor by a small number of rank-one tensors with a high fitness. However, for large scale data, due to necessity to compute Khatri-Rao products of long factors, and multiplication of large matrices, existing algorithms require high computational cost and large memory. Hence decomposition of large-scale tensor is still a challenging problem for PARAFAC. In this paper, we propose a new algorithm based on the ALS algorithm which computes Hadamard products and small matrices, instead of Khatri-Rao products. The new algorithm is able to process extremely large-scale tensor with billions of entries in parallel. Extensive experiments confirm the validity and high performance of the developed algorithm in comparison with other well-known algorithms.

1 Introduction

PARAFAC [3] can be formulated as follows [4] “Factorize a given N -th order tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ into a set of N component matrices: $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \mathbf{a}_2^{(n)}, \dots, \mathbf{a}_J^{(n)}] \in \mathbb{R}^{I_n \times J}$, ($n = 1, 2, \dots, N$) representing the common (loading) factors”, that is,

$$\begin{aligned} \underline{\mathbf{Y}} &\approx \sum_{j=1}^J \mathbf{a}_j^{(1)} \circ \mathbf{a}_j^{(2)} \circ \dots \circ \mathbf{a}_j^{(N)} \\ &= [\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] = [\{\mathbf{A}\}] = \widehat{\underline{\mathbf{Y}}} \end{aligned} \quad (1)$$

with unit-length components $\|\mathbf{a}_j^{(n)}\|_p = 1$ for $n = 1, 2, \dots, N-1$, $j = 1, 2, \dots, J$, and $p = 1, 2$ (see Fig. [5]). Tensor $\widehat{\underline{\mathbf{Y}}}$ is an approximation of the data tensor $\underline{\mathbf{Y}}$.

* Also from Dept. EE Warsaw University of Technology and Systems Research Institute, Polish Academy of Science, Poland

¹ For convenience, tensor notations used in this paper are adopted from [2].

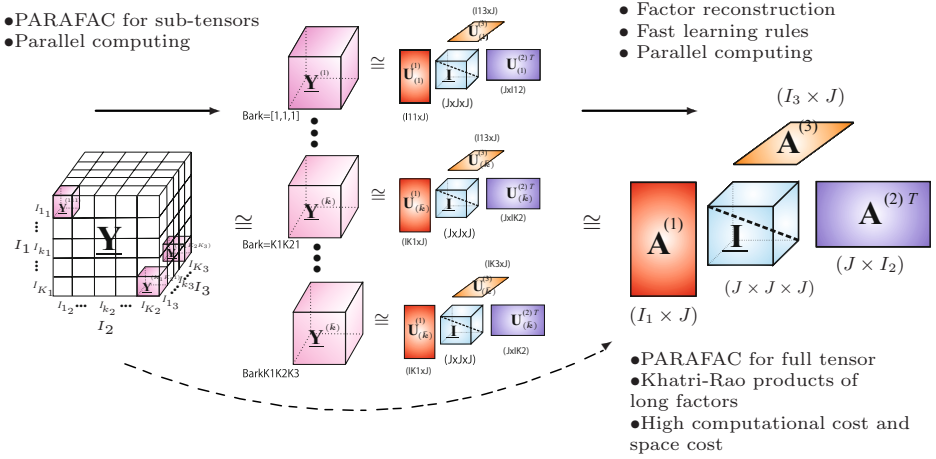


Fig. 1. Illustration for the standard PARAFAC (dash arrow), and grid PARAFAC for large-scale tensors (solid arrows) in two stages

The well-known PARAFAC algorithm is the Alternating Least Squares (ALS) algorithm [2] which minimizes the squared Euclidean distance (Frobenius norm)

$$D(\mathbf{Y} \parallel \hat{\mathbf{Y}}) = \frac{1}{2} \|\mathbf{Y} - \hat{\mathbf{Y}}\|_F^2 \quad (2)$$

with its learning rule for factor $\mathbf{A}^{(n)}$ given by

$$\mathbf{A}^{(n)} \leftarrow \mathbf{Y}_{(n)} \{\mathbf{A}\}^{\odot -n} \left(\{\mathbf{A}^T \mathbf{A}\}^{\otimes -n} \right)^{-1}, \quad (n = 1, 2, \dots, N). \quad (3)$$

where $\{\mathbf{A}\}^{\odot}$ and $\{\mathbf{A}^T \mathbf{A}\}^{\otimes}$ respectively denote Khatri-Rao and Hadamard products of all the matrices inside the curly brackets, whereas $\{\mathbf{A}\}^{\odot -n}$ and $\{\mathbf{A}^T \mathbf{A}\}^{\otimes -n}$ are also products but except the n -th factor.

The mode- n matricized version $\mathbf{Y}_{(n)}$ is an $I_n \times \left(\prod_{k \neq n} I_k\right)$ matrix, and the Khatri-Rao product $\{\mathbf{A}\}^{\odot -n}$ returns a tall matrix of size $\left(\prod_{k \neq n} I_k\right) \times J$ in each iteration step. Hence, for large scale tensor, this learning rule demands high computational cost, and large memory.² The ALS algorithm is relatively slow, and it is impossible to process a dense tensor having billions of entries. To deal with such large dataset, we can reduce number of columns in $\mathbf{Y}_{(n)}$ by some sampled vectors (tubes) along each mode- n which satisfy specific criteria [4, 11]. Recently, the CUR decomposition [5] gives a fast approximation for a raw matrix based on some sampled rows, columns, and their intersection. This method was also extended to tensor to select tubes along each modes. However, both block-wise and CUR approaches have not completely resolved the very large scale

² For a symmetric tensor ($I_1 = \dots = I_N = I$), the computational cost and space cost of the ALS algorithm are $O(JI^N)$, and $O(2I^N)$.

problem for long factors, and high computational cost due to computation of Khatri-Rao product. In this paper, we present a new factorization scheme dealing with large-scale tensor and suitable for parallel implementation. We divide the tensor \mathbf{Y} into a grid of multiple sub-tensors $\mathbf{Y}^{(\bar{\mathbf{k}})}$ of size $I_{k_1} \times I_{k_2} \times \dots \times I_{k_N}$, $\sum_{k_n=1}^{K_n} I_{k_n} = I_n$, where vector $\bar{\mathbf{k}} = [k_1, k_2, \dots, k_N]$ indicates sub-tensor index, $1 \leq k_n \leq K_n$, and K_n is the number of subtensors along the mode- n (see Fig. 1). We factorize all the subtensors to give sub-factors $\mathbf{U}^{(n)}_{(\bar{\mathbf{k}})}$, then estimate the full factors $\mathbf{A}^{(n)}$ for the whole tensor using fast learning rules via parallel computing.

This model is called the grid-tensor factorization (gTF). The proposed algorithm calculates Hadamard products, multiplication of small matrices, and avoids Khatri-Rao products. Especially, this new algorithm opens new perspectives to find nonnegative factors for the very large-scale tensors that could be useful in many applications, such as neural science, data mining. Extensive experiments confirm the validity and high performance of the developed algorithm.

2 ALS Algorithm for Grid PARAFAC

Assuming that N factors $\mathbf{A}^{(n)}$ can explain the tensor \mathbf{Y} , sub-tensors $\mathbf{Y}^{(\bar{\mathbf{k}})}$ can also be factorized by a set of N sub-factors $\{\mathbf{A}_{(\bar{\mathbf{k}})}\} = \{\mathbf{A}_{(k_1)}^{(1)}, \mathbf{A}_{(k_2)}^{(2)}, \dots, \mathbf{A}_{(k_N)}^{(N)}\}$: $\mathbf{Y}^{(\bar{\mathbf{k}})} = \llbracket \{\mathbf{A}_{(\bar{\mathbf{k}})}\} \rrbracket$, where factor $\mathbf{A}^{(n)}$ comprises K_n sub-factors $\mathbf{A}_{(k_n)}^{(n)} \in \mathbb{R}^{I_{k_n} \times J}$: $\mathbf{A}^{(n)} = [\mathbf{A}_{(k_n)}^{(n)T}]_{k_n=1, \dots, K_n}^T$.

The ALS algorithm for grid PARAFAC minimizes the standard Euclidean distance for all the sub-tensors

$$\begin{aligned}
 D &= \frac{1}{2} \sum_{k_1=1}^{K_1} \dots \sum_{k_N=1}^{K_N} \|\mathbf{Y}^{(\bar{\mathbf{k}})} - \llbracket \{\mathbf{A}_{(\bar{\mathbf{k}})}\} \rrbracket\|_F^2 \\
 &= \frac{1}{2} \sum_{\bar{\mathbf{k}}} \|\mathbf{Y}_{(n)}^{(\bar{\mathbf{k}})} - \mathbf{A}_{(k_n)}^{(n)} \{\mathbf{A}_{(\bar{\mathbf{k}})}\}^{\odot -n T}\|_F^2
 \end{aligned} \tag{4}$$

whose gradient components with respect to sub-factors $\mathbf{A}_{(k_n)}^{(n)}$ are given by

$$\begin{aligned}
 \nabla_{\mathbf{A}_{(k_n)}^{(n)}} D &= \sum_{\substack{k_1, \dots, k_{n-1}, \\ k_{n+1}, \dots, k_N}} \left(-\mathbf{Y}_{(n)}^{(\bar{\mathbf{k}})} \mathbf{A}_{(\bar{\mathbf{k}})}^{\odot -n} + \mathbf{A}_{(k_n)}^{(n)} \mathbf{A}_{(\bar{\mathbf{k}})}^{\odot -n T} \mathbf{A}_{(\bar{\mathbf{k}})}^{\odot -n} \right) \\
 &= \sum_{\substack{k_1, \dots, k_{n-1}, \\ k_{n+1}, \dots, k_N}} \left(-\mathbf{Y}_{(n)}^{(\bar{\mathbf{k}})} \mathbf{A}_{(\bar{\mathbf{k}})}^{\odot -n} + \mathbf{A}_{(k_n)}^{(n)} \left\{ \mathbf{A}_{(\bar{\mathbf{k}})}^T \mathbf{A}_{(\bar{\mathbf{k}})} \right\}^{\otimes -n} \right).
 \end{aligned} \tag{5}$$

This leads to the learning rule for sub-factor $\mathbf{A}_{(k_n)}^{(n)}$

$$\mathbf{A}_{(k_n)}^{(n)} \leftarrow \left(\sum_{\substack{k_1, \dots, k_{n-1}, \\ k_{n+1}, \dots, k_N}} \mathbf{Y}_{(n)}^{(\bar{\mathbf{k}})} \mathbf{A}_{(\bar{\mathbf{k}})}^{\odot -n} \right) \left(\sum_{\substack{k_1, \dots, k_{n-1}, \\ k_{n+1}, \dots, k_N}} \left(\mathbf{A}_{(\bar{\mathbf{k}})}^T \mathbf{A}_{(\bar{\mathbf{k}})} \right)^{\otimes -n} \right)^{-1}. \tag{6}$$

Due to relatively small sizes of subtensors, $\mathbf{Y}_{(n)}^{(\bar{k})} \mathbf{A}_{(\bar{k})}^{\odot-n}$, $\left(\mathbf{A}_{(\bar{k})}^T \mathbf{A}_{(\bar{k})}\right)^{\otimes-n}$ can be quickly calculated on parallel workers (labs) or sequentially on a single computer³. Moreover, we can eliminate the sub-tensors involving in estimation of sub-factors $\mathbf{A}_{(k_n)}^{(n)}$ to those built up from tubes sampled by CUR decomposition.

The next section presents optimized algorithm which avoids Khatri-Rao products $\mathbf{A}_{(\bar{k})}^{\odot-n}$ in (6).

3 Optimized ALS Learning Rules

For sub-tensor $\underline{\mathbf{Y}}^{(\bar{k})}$, we factorize this tensor using the ALS algorithm (3) for PARAFAC with $J_{\bar{k}}$ components

$$\underline{\mathbf{Y}}^{(\bar{k})} \approx \llbracket \mathbf{U}_{(\bar{k})}^{(1)}, \mathbf{U}_{(\bar{k})}^{(2)}, \dots, \mathbf{U}_{(\bar{k})}^{(N)} \rrbracket. \tag{7}$$

The number of rank-one tensors $J_{\bar{k}}$ should be chosen so that factors $\mathbf{U}_{(\bar{k})}^{(n)}$ explain as much as possible the sub-tensor $\underline{\mathbf{Y}}^{(\bar{k})}$. Because sub-tensor $\underline{\mathbf{Y}}^{(\bar{k})}$ has small-size, this factorization can easily achieve high fitness. For a subtensor, we have

$$\begin{aligned} \mathbf{Y}_{(n)}^{(\bar{k})} \mathbf{A}_{(\bar{k})}^{\odot-n} &\approx \mathbf{U}_{(\bar{k})}^{(n)} \mathbf{U}_{(\bar{k})}^{\odot-nT} \mathbf{A}_{(\bar{k})}^{\odot-n} = \mathbf{U}_{(\bar{k})}^{(n)} \left(\mathbf{U}_{(\bar{k})}^T \mathbf{A}_{(\bar{k})}\right)^{\otimes-n} \\ &= \mathbf{U}_{(\bar{k})}^{(n)} \left(\mathbf{P}_{(\bar{k})} \circ \left(\mathbf{U}_{(\bar{k})}^{(n)T} \mathbf{A}_{(k_n)}^{(n)}\right)\right), \end{aligned} \tag{8}$$

where $\mathbf{P}_{(\bar{k})} = \left(\mathbf{U}_{(\bar{k})}^T \mathbf{A}_{(\bar{k})}\right)^{\otimes} \in \mathbb{R}^{J_{\bar{k}} \times J}$. Let $\mathbf{Q}_{(\bar{k})} = \left(\mathbf{A}_{(\bar{k})}^T \mathbf{A}_{(\bar{k})}\right)^{\otimes} \in \mathbb{R}^{J \times J}$, from (6), and (8), we obtain the fast update rule for sub-factors $\mathbf{A}_{(k_n)}^{(n)}$

$$\mathbf{A}_{(k_n)}^{(n)} \leftarrow \left(\sum_{\substack{k_1, \dots, k_{n-1}, \\ k_{n+1}, \dots, k_N}} \mathbf{U}_{(\bar{k})}^{(n)} \frac{\mathbf{P}_{(\bar{k})}}{\mathbf{U}_{(\bar{k})}^{(n)T} \mathbf{A}_{(k_n)}^{(n)}} \right) \left(\sum_{\substack{k_1, \dots, k_{n-1}, \\ k_{n+1}, \dots, k_N}} \frac{\mathbf{Q}_{(\bar{k})}}{\mathbf{A}_{(k_n)}^{(n)T} \mathbf{A}_{(k_n)}^{(n)}} \right)^{-1}. \tag{9}$$

The expression (9) calculates Hardamard products, and performs all operations on small-sized matrices, instead of Khatri-Rao products for long matrices. Matrices $\mathbf{P}_{(\bar{k})} \in \mathbb{R}^{J_{\bar{k}} \times J}$ and $\mathbf{Q}_{(\bar{k})} \in \mathbb{R}^{J \times J}$ can be calculated only once time, and can be quickly updated after estimating sub-factors $\mathbf{A}_{(k_n)}^{(n)}$. For a symmetric tensor, the complexity of (9) is $O(3J^2 K^{N-1} I/L)$, where L is the number of labs in a parallel system. The pseudo-code of the new ALS algorithm is given in Algorithm 1⁴. Parallel FOR-loop denoted by “**parfor**” loop is available with the Matlab Parallel Computing Toolbox.

³ Using block multiplication $[\mathbf{Y}_1 \ \mathbf{Y}_2][\mathbf{A}_1; \ \mathbf{A}_2] = \mathbf{Y}_1 \mathbf{A}_1 + \mathbf{Y}_2 \mathbf{A}_2$, the learning rule (6) can be directly derived from (3).

⁴ The normalization of components to unit-length vectors is not explicitly displayed in Algorithm 1.

Algorithm 1. Fast ALS for large scale PARAFAC

```

1 begin
2   initialization  $\mathbf{A}_{(k_n)}^{(n)}, \forall n, \forall k_n$ 
3   parfor sub-tensor  $\mathbf{Y}^{(\bar{k})}$  do
4      $[\mathbf{U}_{(\bar{k})}^{(1)}, \dots, \mathbf{U}_{(\bar{k})}^{(N)}] = \text{parafacALS}(\mathbf{Y}^{(\bar{k})}, J_{\bar{k}})$ 
5      $\mathbf{P}_{(\bar{k})} = \prod_{n=1}^N (\mathbf{U}_{(\bar{k})}^{(n)T} \mathbf{A}_{(k_n)}^{(n)})$ ,  $\mathbf{Q}_{(\bar{k})} = \prod_{n=1}^N (\mathbf{A}_{(k_n)}^{(n)T} \mathbf{A}_{(k_n)}^{(n)})$ 
6   endfor
7   repeat
8     for  $n = 1$  to  $N$  do
9       foreach  $k_n = 1$  to  $K_n$  do
10         $\mathbf{T} = \mathbf{0}$ ,  $\mathbf{S} = \mathbf{0}$ 
11        parfor  $[\bar{k}]_n = k_n$  do
12           $\mathbf{P}_{(\bar{k})} = \mathbf{P}_{(\bar{k})} \oslash (\mathbf{U}_{(\bar{k})}^{(n)T} \mathbf{A}_{(k_n)}^{(n)})$ ,  $\mathbf{T} = \mathbf{T} + \mathbf{U}_{(\bar{k})}^{(n)} \mathbf{P}_{(\bar{k})}$ 
13           $\mathbf{Q}_{(\bar{k})} = \mathbf{Q}_{(\bar{k})} \oslash (\mathbf{A}_{(k_n)}^{(n)T} \mathbf{A}_{(k_n)}^{(n)})$ ,  $\mathbf{S} = \mathbf{S} + \mathbf{Q}_{(\bar{k})}$ 
14        endfor
15         $\mathbf{A}_{(k_n)}^{(n)} \leftarrow \mathbf{T} \mathbf{S}^{-1}$  /* Update  $\mathbf{A}_{(k_n)}^{(n)}$  */
16      end
17      parfor each  $\bar{k}$  do
18         $\mathbf{P}_{(\bar{k})} = \mathbf{P}_{(\bar{k})} \otimes (\mathbf{U}_{(\bar{k})}^{(n)T} \mathbf{A}_{(k_n)}^{(n)})$ ,  $\mathbf{Q}_{(\bar{k})} = \mathbf{Q}_{(\bar{k})} \otimes (\mathbf{A}_{(k_n)}^{(n)T} \mathbf{A}_{(k_n)}^{(n)})$ 
19      endfor
20    end
21  until a stopping criterion is met
22 end

```

4 Stopping Criterion

Stopping criterion takes an important role in identification of convergence of a factorization. For simplicity, the cost function value (2) is usually used as stopping criterion [5]. However, for a large tensor, an explicit computation of the cost function value (2) is impossible due to so much memory requirement to build up the approximate tensor $\hat{\mathbf{Y}}$. In this section, we derive a fast computation for stopping criterion applied to the grid PARAFAC. The Frobenius norm of a raw sub-tensor and its approximation is given by

$$D(\bar{k}) = \|\mathbf{Y}^{(\bar{k})} - \hat{\mathbf{Y}}^{(\bar{k})}\|_F^2 = \|\mathbf{Y}^{(\bar{k})}\|_F^2 + \|\hat{\mathbf{Y}}^{(\bar{k})}\|_F^2 - 2 \langle \mathbf{Y}^{(\bar{k})}, \hat{\mathbf{Y}}^{(\bar{k})} \rangle \quad (10)$$

where $\langle \mathbf{Y}, \hat{\mathbf{Y}} \rangle$ is the inner product of two same-sized tensors

$$\langle \mathbf{Y}^{(\bar{k})}, \hat{\mathbf{Y}}^{(\bar{k})} \rangle = \sum_{i_1=1}^{I_1} \dots \sum_{i_N=1}^{I_N} y_{i_1 \dots i_N}^{(\bar{k})} \hat{y}_{i_1 \dots i_N}^{(\bar{k})} = \text{vec}(\mathbf{Y}_{(N)}^{(\bar{k})})^T \text{vec}(\hat{\mathbf{Y}}_{(N)}^{(\bar{k})}). \quad (11)$$

Each terms in the expression (10) can be computed as follows

$$\|\hat{\mathbf{Y}}^{(\bar{k})}\|_F^2 = \mathbf{1}^T \{ \mathbf{A}_{(\bar{k})}^T \mathbf{A}_{(\bar{k})} \}^* \mathbf{1} = \mathbf{1}^T \mathbf{Q}_{(\bar{k})} \mathbf{1}, \quad (12)$$

⁵ FIT rate (FIT(%)) = $1 - \frac{\|\mathbf{Y} - \hat{\mathbf{Y}}\|_F^2}{\|\mathbf{Y}\|_F^2}$ can also be used. However, due to similar computation, we only mention the Frobenius norm.

$$\begin{aligned} \langle \underline{\mathbf{Y}}^{(\bar{k})}, \widehat{\underline{\mathbf{Y}}}^{(\bar{k})} \rangle &= \mathbf{1}^T \{ \underline{\mathbf{U}}^{(\bar{k})} \}^{\odot T} \{ \underline{\mathbf{A}}^{(\bar{k})} \}^{\odot} \mathbf{1} \\ &= \mathbf{1}^T \left\{ \underline{\mathbf{U}}^{(\bar{k})T} \underline{\mathbf{A}}^{(\bar{k})} \right\}^{\circledast} \mathbf{1} = \mathbf{1}^T \underline{\mathbf{P}}^{(\bar{k})} \mathbf{1}. \end{aligned} \quad (13)$$

From (10), (12) and (13), we obtain a convenient and fast computing for the cost function

$$\begin{aligned} D &= \frac{1}{2} \sum_{\bar{k}} D^{(\bar{k})} = \frac{1}{2} \sum_{\bar{k}} \left(\|\underline{\mathbf{Y}}^{(\bar{k})}\|_F^2 + \mathbf{1}^T \underline{\mathbf{Q}}^{(\bar{k})} \mathbf{1} - 2 \mathbf{1}^T \underline{\mathbf{P}}^{(\bar{k})} \mathbf{1} \right) \\ &= \frac{1}{2} \|\underline{\mathbf{Y}}\|_F^2 + \frac{1}{2} \sum_{\bar{k}} \left(\mathbf{1}^T \underline{\mathbf{Q}}^{(\bar{k})} \mathbf{1} - 2 \mathbf{1}^T \underline{\mathbf{P}}^{(\bar{k})} \mathbf{1} \right). \end{aligned} \quad (14)$$

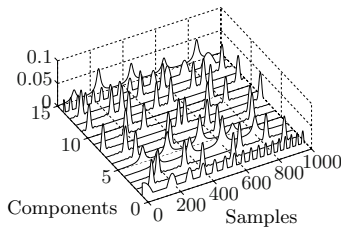
The first term $\|\underline{\mathbf{Y}}\|_F^2$ is constant, hence can be neglected. The rest terms are additions of all the entries of matrices $\underline{\mathbf{Q}}^{(\bar{k})}$, and $\underline{\mathbf{P}}^{(\bar{k})}$.

5 Experiments

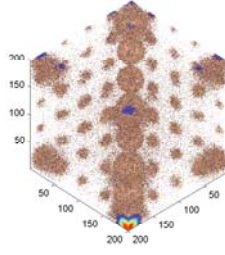
5.1 Synthetic Benchmark

In the first example, we factorized a synthetic tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{5000 \times 5000 \times 5000}$ built up from 15 nonnegative components (shown in Fig. 2(a)), and next degraded by an additive Gaussian noise at SNR = 0 dB⁶. A partial data with 200 samples along each dimension is illustrated in Fig. 2(b), the 45-th slice of this noisy tensor is also given in Fig. 3(a). This dense tensor with 125 billions of entries could consume 500 GB of memory. Factorizing such tensor using existing PARAFAC algorithms is impossible because of large tensor size. However, the proposed algorithm can quickly deal with this problem. In the approximate step, we divided this tensor into 8000 sub-tensors of size $250 \times 250 \times 250$, and simultaneously factorized them with $J_a = 25$ PARAFAC components in a parallel system with 16 labs to obtain 8000 sub-factors $\underline{\mathbf{U}}^{(n)}_{(\bar{k})} \in \mathbb{R}^{250 \times 25}$, $n = 1, 2, 3, \bar{k} = [k_1, k_2, k_3], k_n = 1, \dots, 20$. This step took 3680 seconds. The experiment was run on MATLAB ver 2008b and its Distributed Computing Server and Parallel Computing toolboxes. The full factors were estimated in two stages to reduce inter-communication between labs. In the first stage, 16 groups of 500 consecutive sub-factors in sub-tensors of size $1250 \times 1250 \times 5000$ were used to simultaneously estimate 16 sets of sub-factors. Then from these sub-factors, we built up the full factors for tensor $5000 \times 5000 \times 5000$. The whole step 2 took 56.51 seconds. With these estimated PARAFAC factors, we can quickly retrieve the nonnegative factors under the data by applying the fast LS algorithm [6]. This step only took 2.78 seconds. The 15 estimated factors achieved high SIR indices in a range of [43.64, 54.97] dB, and are depicted in Fig. 2(a). Fig. 3(b) is the reconstruction of the noisy slice in Fig. 3(a).

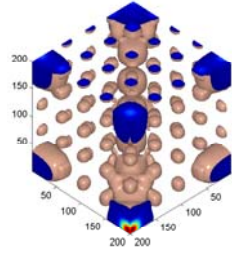
⁶ The standard deviation of noise is given by $\sigma_n^2 = \mathbb{E}[\underline{\mathbf{Y}}^2] = \mathbf{1}^T \{ \underline{\mathbf{A}}^T \underline{\mathbf{A}} \}^{\circledast} \mathbf{1} / \prod I_n$.



(a) 15 components

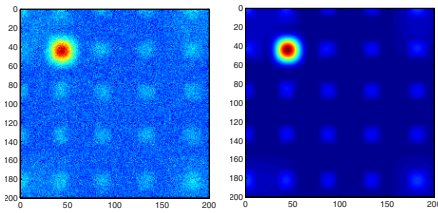


(b) Noisy tensor



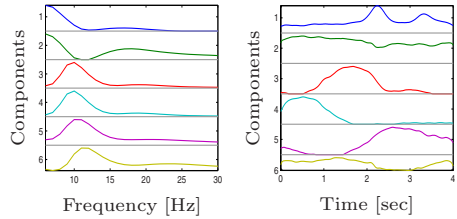
(c) Reconstructed tensor

Fig. 2. Illustration for Example 1 with a noisy dense tensor of size $5000 \times 5000 \times 5000$ (only 200 samples for each dimensions are shown)



(a) Noisy slice

(b) Reconstructed slice



(a) Spectral components

(b) Temporal components

Fig. 3. Noisy slice and its reconstruction for Example 1 (only 200 samples for each dimensions are shown)

Fig. 4. Illustration of 6 nonnegative components estimated from 10 PARAFAC components for the Graz benchmark

5.2 Graz EEG Dataset for BCI

The Graz dataset [7] contains EEG signals involving left hand, right hand, foot, tongue imagery movements acquired from 60 channels in a duration of 7 seconds (4 seconds after trigger). The dataset was recorded from 3 subjects, and had 840 trials. All the EEG signals were transformed into the time-frequency domain using the complex Morlet wavelet, to have a spectral tensor $60 \text{ channels} \times 25 \text{ frequency bins (6-30 Hz)} \times 250 \text{ time frames} \times 840 \text{ trials}$. Due to meaningful factorization, the hidden factors under this EEG spectral tensor require nonnegative constraints, and are considered as useful features for successful EEG classification [8]. Therefore, we firstly estimated PARAFAC factors of this tensor, then extracted nonnegative factors from them using the fast LS algorithm [6]. This dense tensor had a total of 315 millions of entries, and consumed 1.26 GB of memory. Factorization of this full tensor with 10 components took 3900 seconds on a quad core computer (2.67 GHz, 8 GB memory), and achieved FIT = 78.95%. However, the grid ALS algorithm for a grid of 16 sub-tensors divided from the EEG tensor along the 4-th dimension (trials) only took 112 seconds

to extract the same number components with $\text{FIT} = 78.55\%$. The nonnegative factors quickly derived from both approaches only took 0.41 seconds, and respectively explain 77.08% and 77.07% of the raw tensor for the full and grid processing. The components of the spectral and temporal factors are shown in Fig. 4. The 3 factors $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)}$, $\mathbf{A}^{(3)}$ can be used as bases to extract feature in EEG classification which is out of scope of this paper due to space limit.

6 Conclusions

We present the new fast and robust ALS algorithm for large-scale PARAFAC. The validity and high performance of the proposed algorithm have been confirmed even for noisy data, and also for the large scale BCI benchmark. The new fast stopping criterion is proposed for this algorithm. Variations of the ALS algorithm for PARAFAC with regularized terms such as total variation, sparsity, smoothness, nonnegativity, orthogonality constraints can be applied to the grid PARAFAC with some modifications on the learning rule (9). Strategy for grid division of a tensor can affect to the performance of factorization, and the running time of parallel computing. Basically, sub-tensors' sizes should satisfy unique conditions of PARAFAC [2]. Moreover, sub-tensor should have maximum possible number of entries in its working lab. The total data transferred between client and all labs is briefed here as: $2J \sum_n I_n + N J^2 \prod_n K_n$. This means that it is better to have a minimum number of sub-tensors. Finally, inter-communication between labs should be limited as much as possible. To deal with this, we can estimate the full factors in multistage as illustrated in Example 1. Due to the page limit, we presented briefly some discussion points for grid tensor factorization and its ALS algorithm.

References

1. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.: Nonnegative Matrix and Tensor Factorizations. Wiley, Chichester (2009)
2. Kolda, T., Bader, B.: Tensor decompositions and applications. *SIAM Review* 51(3) (in print, September 2009)
3. Harshman, R.: Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis. *UCLA Working Papers in Phonetics* 16, 1–84 (1970)
4. Cichocki, A., Phan, A.H.: Fast local algorithms for large scale nonnegative matrix and tensor factorizations. In: *IEICE (invited paper)* (March 2009)
5. Goreinov, S., Tyrtshnikov, E., Zamarashkin, N.: A theory of pseudoskeleton approximations. *Linear Algebra and Applications* 261, 1–21 (1997)
6. Phan, A.H., Cichocki, A.: Fast nonnegative tensor factorization for very large-scale problems using two-stage procedure. In: *CAMSAP* (2009)
7. Brunner, C., Leeb, R., Müller-putz, G.R., Schlögl, A., Pfurtscheller, G.: BCI competition 2008. Graz data set A (2009)
8. Mørup, M., Hansen, L., Parnas, J., Arnfred, S.: Decomposing the time-frequency representation of EEG using non-negative matrix and multi-way factorization. Technical report (2006)

An Observation Angle Dependent Nonstationary Covariance Function for Gaussian Process Regression

Arman Melkumyan and Eric Nettleton

Australian Centre for Field Robotics, The University of Sydney, NSW 2006, Australia
{a.melkumyan, e.nettleton}@acfr.usyd.edu.au

Abstract. Despite the success of Gaussian Processes (GPs) in machine learning, the range of applications and expressiveness of GP models are confined by the limited set of available covariance functions. This paper presents a new non-stationary covariance function which allows simple geometric interpretation and depends on the angle at which points can be seen from an observation centre. The construction of the new covariance function and the proof of its positive semi-definiteness are based on geometric reasoning combined with analytic computations. Experiments conducted with both artificial and real datasets demonstrate the advantages of the developed covariance function.

Keywords: Gaussian Process, covariance function, non-stationary stochastic process, Mercer kernel.

1 Introduction and Related Work

During the past decade Gaussian Processes (GPs) have been successfully employed for regression in supervised machine learning. The range of applications includes geophysics, mining, hydrology, finances, reservoir engineering and robotics. However, as learning with Gaussian Processes is equivalent to identifying correlations between points, the predictive qualities of GP models fully depend on the choice of the covariance functions (kernels). The set of already developed kernels is quite limited and Rasmussen and Williams [1] suggested that an important area of future developments for GP models is the construction and use of more sophisticated and expressive covariance functions.

Although the properties of the known kernels can be combined by considering their products and weighted sums, development of new covariance functions that increase the predictive quality and expressiveness of GPs is a challenging task.

Williams [2] derived a covariance function for GPs corresponding to neural networks with sigmoidal and Gaussian hidden units. It supports efficient predictions using GPs for neural networks with an infinite number of hidden units.

Sugiyama et al. [3] proposed Gaussian kernels which are defined on the non-linear manifolds for value function approximation. These kernels are smooth along the graph, robust against the graph estimation error and easy to compute.

Melkumyan and Ramos [4] recently developed the Sparse covariance function which naturally provides sparse covariance matrices and enables exact GP inference even for large datasets, providing both storage and computational benefits.

In the present work a new non-stationary covariance function is developed based on geometric reasoning and closed form calculation of integrals. The new covariance function depends on the angle at which points can be observed from an observation centre and therefore is named the Observation Angle Dependent (OAD) covariance function. The numerical evaluations presented in the experiment section show that the predictive qualities of the OAD covariance function compare favorably with the predictive qualities of other popular covariance functions.

This paper is organized as follows. Section 2 reviews the basics of GP regression and introduces notation. In Section 3 the new Observation Angle Dependent (OAD) covariance function and its main properties are derived. Section 4 presents the partial derivatives for learning. The predictive quality of the OAD covariance function is evaluated in Section 5 via experiments on both artificial and real datasets. Finally, Section 6 concludes the paper and discusses further developments.

2 Gaussian Processes

This section briefly reviews GPs and introduces notation. Detailed information on different aspects of Gaussian processes for machine learning is available in [1]. Consider the supervised learning problem with a training set $D = (x_i, y_i), i = 1 : N$, consisting of N input points $\mathbf{x}_i \in \mathbb{R}^D$ and the corresponding outputs $y_i \in \mathbb{R}$. The objective is to compute the predictive distribution $f(\mathbf{x}_*)$ at a new test point \mathbf{x}_* . A GP model places a multivariate Gaussian distribution over the space of function variables $f(\mathbf{x})$ mapping input to output spaces. The model is specified by defining a mean function $m(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$ of the GP. Assuming Gaussian noise ε with variance σ^2 in observations, so that $y = f(x) + \varepsilon$, and denoting $(X, \mathbf{f}, y) = (\{\mathbf{x}_i\}, \{f_i\}, \{y_i\})_{i=1:N}$, $(X_*, \mathbf{f}_*, y_*) = (\{\mathbf{x}_{*i}\}, \{f_{*i}\}, \{y_{*i}\})_{i=1:N}$ for the training and testing sets respectively, the joint distribution with $m(\mathbf{x}) = 0$ becomes

$$\begin{bmatrix} y \\ \mathbf{f}_* \end{bmatrix} \sim N \left(0, \begin{bmatrix} K(X, X) + \sigma^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix} \right). \tag{1}$$

Here $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and K is the covariance matrix computed between all points in the set.

By conditioning on the observed training points, the predictive distribution for new points can be obtained as $p(f_* | X_*, X, \mathbf{y}) = N(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$ where

$$\begin{aligned} \boldsymbol{\mu}_* &= K(X_*, X) [K(X, X) + \sigma^2 I]^{-1} \mathbf{y}, \\ \boldsymbol{\Sigma}_* &= K(X_*, X_*) - K(X_*, X) [K(X, X) + \sigma^2 I]^{-1} K(X, X_*) + \sigma^2 I. \end{aligned} \tag{2}$$

Learning a GP model is equivalent to determining the hyper-parameters of the covariance function from some training dataset. In a Bayesian framework this can be performed by maximizing the log of the marginal likelihood (lml) w.r.t. θ :

$$\log p(\mathbf{y} | X, \theta) = -\frac{1}{2} \mathbf{y}^T [K(X, X) + \sigma^2 I]^{-1} \mathbf{y} - \frac{1}{2} \log |K(X, X) + \sigma^2 I| - \frac{N}{2} \log 2\pi \tag{3}$$

Eq. (3) has three terms (from left to right) representing the data fit, complexity penalty (encoding the Occam’s Razor principle) and a normalization constant. It is a non-convex function on the hyper-parameters and its local maxima can be obtained with gradient descent techniques by using multiple starting points. However, this requires the computation of partial derivatives of lml resulting in:

$$\frac{\partial}{\partial \theta_j} \log p(\mathbf{y} | X, \theta) = \frac{1}{2} \mathbf{y}^T K^{-1} \frac{\partial K}{\partial \theta_j} K^{-1} \mathbf{y} - \frac{1}{2} \text{tr} \left(K^{-1} \frac{\partial K}{\partial \theta} \right). \tag{4}$$

which requires computation of partial derivatives of the covariance function w.r.t. θ .

3 Observation Angle Dependent (OAD) Covariance Function

The desired OAD covariance function must depend only on the angle at which points can be observed from an observation center \mathbf{x}_c . As that observation angle depends not on the difference $\mathbf{x} - \mathbf{x}'$ but on the spatial location of the points \mathbf{x} and \mathbf{x}' with respect to \mathbf{x}_c , the resulting covariance function will be non-stationary.

The OAD covariance function will be first constructed in isotropic form in the case of two dimensions as this is the most convenient case for visual demonstration of the main ideas. Then it will be extended to arbitrary dimensions and made anisotropic. A proof of positive semi-definiteness will be provided for the most general case.

3.1 Construction of the OAD Covariance Function

Consider the piecewise constant transfer function

$$h(\mathbf{x}, \mathbf{u}; \mathbf{x}_c) = \begin{cases} a_0, & \text{if } \alpha(\mathbf{x}, \mathbf{u}; \mathbf{x}_c) < \pi/2 \\ b_0, & \text{if } \alpha(\mathbf{x}, \mathbf{u}; \mathbf{x}_c) > \pi/2 \end{cases}, \tag{5}$$

where $\alpha(\mathbf{x}, \mathbf{u}; \mathbf{x}_c)$ represents the angle between the points \mathbf{x} and \mathbf{u} as seen from the observation centre \mathbf{x}_c . Conducting derivations analogous to those presented in [1] and using the transfer function $h(\mathbf{x}, \mathbf{u}; \mathbf{x}_c)$ the following covariance function is obtained:

$$k_2(\mathbf{x}, \mathbf{x}') = \sigma^2 \int_{\|\mathbf{u} - \mathbf{x}_c\|_2 = 1} h(\mathbf{x}, \mathbf{u}; \mathbf{x}_c) h(\mathbf{x}', \mathbf{u}; \mathbf{x}_c) d\mathbf{u} . \tag{6}$$

Here $\|\cdot\|_2$ is the two dimensional Euclidian norm and the integration is conducted through the circumference of the unit circle with centre \mathbf{x}_c .

The integral in Eq. (6) can be analytically evaluated (see Appendix A) to result in:

$$K(\mathbf{x}, \mathbf{x}'; \mathbf{x}_c) = \sigma_0^2 \left(1 - \frac{1 - \sin \varphi}{\pi} \alpha(\mathbf{x}, \mathbf{x}'; \mathbf{x}_c) \right), \tag{7}$$

where σ_0, φ are scalar hyper-parameters of the covariance function and $\alpha(\mathbf{x}, \mathbf{x}'; \mathbf{x}_c)$ is the angle between the points \mathbf{x} and \mathbf{x}' as seen from the observation centre \mathbf{x}_c .

3.2 Arbitrary Dimensions and Anisotropy

In the case of arbitrary dimensions D the definition of the transfer function Eq. (5) remains unchanged, but now $\alpha(\mathbf{x}, \mathbf{u}; \mathbf{x}_c)$ represents the angle between D dimensional points \mathbf{x} and \mathbf{u} as observed from the D dimensional centre \mathbf{x}_c . The observation angle $\alpha(\mathbf{x}, \mathbf{u}; \mathbf{x}_c)$ can be analytically calculated using the properties of the dot product for vectors:

$$\alpha(\mathbf{x}, \mathbf{u}; \mathbf{x}_c) = \arccos \frac{(\mathbf{x} - \mathbf{x}_c)^T (\mathbf{x}' - \mathbf{x}_c)}{\sqrt{(\mathbf{x} - \mathbf{x}_c)^T (\mathbf{x} - \mathbf{x}_c)} \sqrt{(\mathbf{x}' - \mathbf{x}_c)^T (\mathbf{x}' - \mathbf{x}_c)}}. \tag{8}$$

Definition of the covariance function Eq. (6) in the D dimensional case becomes

$$k_D(\mathbf{x}, \mathbf{x}') = \sigma^2 \int_{\|\mathbf{u} - \mathbf{x}_c\|_D = 1} h(\mathbf{x}, \mathbf{u}; \mathbf{x}_c) h(\mathbf{x}', \mathbf{u}; \mathbf{x}_c) d\mathbf{u}, \tag{9}$$

where $\|\cdot\|_D$ is the D dimensional Euclidian norm and the integration is conducted on the surface of the D dimensional unit sphere with centre \mathbf{x}_c . Using D dimensional spherical coordinate system, calculations analogous to the ones in two dimensional case can be carried out for Eq. (9). The result is again Eq. (7) where $\alpha(\mathbf{x}, \mathbf{u}; \mathbf{x}_c)$ is now the angle between D dimensional vectors and can be calculated using Eq. (8).

The resultant covariance function (7)-(8) can be made anisotropic by applying a non-singular linear transformation to the multi-dimensional space. The transformation will result in replacing the vectors \mathbf{x}, \mathbf{x}' and \mathbf{x}_c by $\mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{x}'$ and $\mathbf{A}\mathbf{x}_c$ where \mathbf{A} is the non-singular transformation matrix. Using Eqs. (7), (8) and the transformation matrix \mathbf{A} , the following multi-dimensional anisotropic form of the OAD covariance function is obtained:

$$K(\mathbf{x}, \mathbf{x}'; \mathbf{x}_c, \varphi, \mathbf{\Omega}) = \sigma_0^2 \left(1 - \frac{1 - \sin \varphi}{\pi} \arccos \frac{(\mathbf{x} - \mathbf{x}_c)^T \mathbf{\Omega} (\mathbf{x}' - \mathbf{x}_c)}{\sqrt{(\mathbf{x} - \mathbf{x}_c)^T \mathbf{\Omega} (\mathbf{x} - \mathbf{x}_c)} \sqrt{(\mathbf{x}' - \mathbf{x}_c)^T \mathbf{\Omega} (\mathbf{x}' - \mathbf{x}_c)}} \right) \tag{10}$$

where $\mathbf{\Omega} = \mathbf{A}^T \mathbf{A}$ is a symmetric positive semi-definite matrix.

The hyper-parameters of the OAD covariance function (10) are the scalars σ_0 and φ , D dimensional vector \mathbf{x}_c and $D \times D$ symmetric positive semi-definite matrix $\mathbf{\Omega}$. The resulting total number of scalar hyper-parameters is equal to $2 + D(D+3)/2$.

If $\mathbf{\Omega}$ is diagonal, it can be expressed via the characteristic length-scales:

$$\mathbf{\Omega} = \text{diag}(l_1^{-2}, l_2^{-2}, \dots, l_D^{-2}) \tag{11}$$

and the number of scalar hyper-parameters in this case decreases to $2(D+1)$.

As the number of hyper-parameters in the fully anisotropic case is greater by $D(D-1)/2$ than in the case of diagonal $\mathbf{\Omega}$, the learning stage of the GP can have high computational cost if high dimensional problems are considered with full matrix $\mathbf{\Omega}$. If information is available about the anisotropic characteristics of the problem, pre-processing the data bringing it into a form suitable for using Eq. (11) can provide significant computational savings for the learning stage.

In Eq. (10) the vectors \mathbf{x} , \mathbf{x}' and \mathbf{x}_c can be replaced by the corresponding augmented vectors $\tilde{\mathbf{x}}=(1, x_1, \dots, x_D)^T$, $\tilde{\mathbf{x}}'=(1, x'_1, \dots, x'_D)^T$ and $\tilde{\mathbf{x}}_c=(0, x_{c,1}, \dots, x_{c,D})^T$ where the first entries correspond to the bias.

Using Eq. (9) and the linear transformation matrix \mathbf{A} , one has that for any points \mathbf{x}_i and any real numbers c_i where $i=1,2,\dots,n$ the inequality

$$\sum_{i,j=1}^n c_i c_j k_D(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \int_{\|\mathbf{u}-\mathbf{Ax}_i\|_0=1} \left(\sum_{i=1}^n c_i h(\mathbf{Ax}_i, \mathbf{u}; \mathbf{Ax}_c) \right)^2 du \geq 0 \tag{12}$$

holds, which proves the positive semi-definiteness of the OAD covariance function.

4 Partial Derivatives for Learning

Learning the GP requires the computation of covariance function’s partial derivatives w.r.t. the hyper-parameters (Eq. (4)). Based on Eq. (10) the following expressions for the partial derivatives of the OAD covariance function can be calculated:

$$\frac{\partial K}{\partial \sigma_0} = \frac{2}{\sigma_0} K, \quad \frac{\partial K}{\partial \varphi} = \frac{\cos \varphi}{1 - \sin \varphi} (\sigma_0^2 - K), \tag{13}$$

$$\begin{aligned} \nabla_{\mathbf{x}_c} K = & \sigma_0^2 \frac{1 - \sin \varphi}{\pi} \mathbf{\Omega} \left[\frac{\mathbf{x} + \mathbf{x}' - 2\mathbf{x}_c}{(\mathbf{x} - \mathbf{x}_c)^T \mathbf{\Omega} (\mathbf{x}' - \mathbf{x}_c)} + \frac{\mathbf{x}_c - \mathbf{x}}{(\mathbf{x} - \mathbf{x}_c)^T \mathbf{\Omega} (\mathbf{x} - \mathbf{x}_c)} \right. \\ & \left. + \frac{\mathbf{x}_c - \mathbf{x}'}{(\mathbf{x}' - \mathbf{x}_c)^T \mathbf{\Omega} (\mathbf{x}' - \mathbf{x}_c)} \right] \cot \left(\frac{\pi}{1 - \sin \varphi} \frac{K - \sigma_0^2}{\sigma_0^2} \right), \tag{14} \end{aligned}$$

$$\begin{aligned} \frac{\partial K}{\partial \Omega_{ij}} = & \frac{\sigma_0^2}{\pi} \frac{1 - \sin \varphi}{1 + \delta_{ij}} \left[\frac{(x_i - x_{c,i})(x_j - x_{c,j})}{(\mathbf{x} - \mathbf{x}_c)^T \mathbf{\Omega} (\mathbf{x} - \mathbf{x}_c)} + \frac{(x'_i - x_{c,i})(x'_j - x_{c,j})}{(\mathbf{x}' - \mathbf{x}_c)^T \mathbf{\Omega} (\mathbf{x}' - \mathbf{x}_c)} \right. \\ & \left. - \frac{(x_i - x_{c,i})(x'_j - x_{c,j}) + (x_j - x_{c,j})(x'_i - x_{c,i})}{(\mathbf{x} - \mathbf{x}_c)^T \mathbf{\Omega} (\mathbf{x}' - \mathbf{x}_c)} \right] \cot \left(\frac{\pi}{1 - \sin \varphi} \frac{K - \sigma_0^2}{\sigma_0^2} \right), \tag{15} \end{aligned}$$

where $\nabla_{\mathbf{x}_c} K = (\partial K / \partial x_{c,1}, \partial K / \partial x_{c,2}, \dots, \partial K / \partial x_{c,D})^T$ and $\delta_{ij} = 1$ if $i = j$, $\delta_{ij} = 0$ if $i \neq j$.

If $\mathbf{\Omega}$ is diagonal, then Eq. (11) with characteristic length-scales can be used and the corresponding partial derivatives can be obtained from Eq. (15):

$$\frac{\partial K}{\partial l_i} = -\sigma_0^2 \frac{1 - \sin \varphi}{\pi l_i} \left[\frac{\left[\frac{(x_i - x_{c,i})}{l_i} \right]^2}{\sum_{k=1}^D \left[\frac{(x_k - x_{c,k})}{l_k} \right]^2} + \frac{\left[\frac{(x'_i - x_{c,i})}{l_i} \right]^2}{\sum_{k=1}^D \left[\frac{(x'_k - x_{c,k})}{l_k} \right]^2} - \frac{2(x_i - x_{c,i})(x'_i - x_{c,i})/l_i^2}{\sum_{k=1}^D (x_k - x_{c,k})(x'_k - x_{c,k})/l_k^2} \right] \cot \left(\frac{\pi}{1 - \sin \varphi} \frac{K - \sigma_0^2}{\sigma_0^2} \right). \tag{16}$$

5 Experiments

This section provides empirical comparisons between the proposed OAD and the popular squared exponential (SqExp), Matérn and neural network (NN) covariance functions [1] using both artificially created and real datasets.

5.1 Artificial Dataset

The dataset for this experiment is constructed by sampling from the step function with Gaussian noise with standard deviation $\sigma_n = 0.1$. SqExp and Matérn covariance functions are both known to be poor models for discontinuous functions [1], therefore only the OAD and NN covariance functions are considered here. It can be observed from Fig. 1 that both the OAD and NN covariance functions model the data correctly if the discontinuity happens at the origin $x=0$. However, the predictions using NN become oscillatory and provide poor model for the data when the point of discontinuity moves away from the origin. The OAD covariance function models the data correctly in all the considered situations.

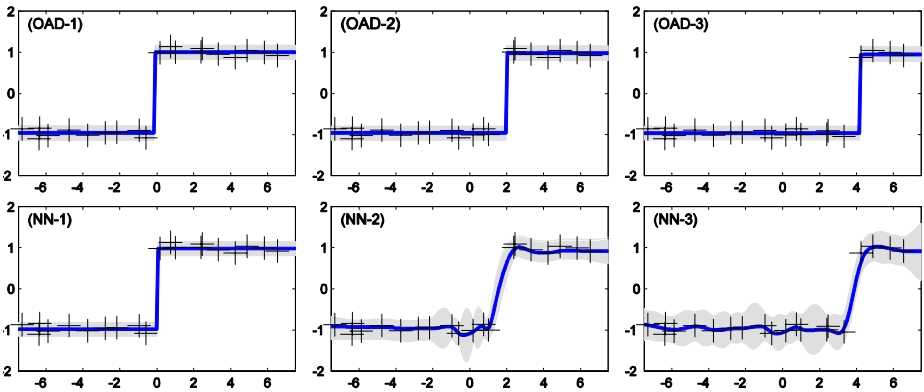


Fig. 1. Modeling discontinuities via the OAD and NN covariance functions. The predictive quality of the NN covariance function becomes poor if the discontinuity happens away from the origin, while the OAD covariance function correctly models all the considered situations.

5.2 Rainfall Dataset

In this experiment the predictive qualities of the OAD, SqExp, Matérn and NN covariance functions are compared on the Spatial Interpolation Comparison dataset [5] (SIC)¹ which is popular in geostatistics for comparing predictive models. The dataset consists of 467 points measuring rainfall in 2D space. The points are divided into two sets, inference and testing. The inference set contains the points used to perform inference on the testing points. For each case the experiment is repeated 2000 times with randomly selected inference and testing sets. Fig. 2a shows the normalized mean squared error (MSE) for the different covariance functions and the standard deviation (one sigma for each part of the bar) as a function of the number of inference points. The results demonstrate that the OAD covariance function systematically leads to better predictions regardless of the chosen inference and testing sets. Fig. 2b shows the percentage of additional MSE that other covariance functions produce compared with the OAD covariance functions. From Fig. 2b it can be observed that in the case of 30 inference points the NN covariance function leads to 35% greater MSE and SqExp, Matérn 3/2 and Matérn 5/2 lead to about 60% greater MSE than the OAD covariance function. When the number of inference points increases, more information becomes available for the GP regression and less sophistication is required from the covariance function to model the data correctly. This is why the percentage of the additional MSE monotonically decreases with the increase of the number of inference points in Fig. 2b. However, even with 200 inference points NN leads to about 13% and SqExp, Matérn 3/2 and Matérn 5/2 lead to about 20% greater mean square error than the OAD covariance function does.

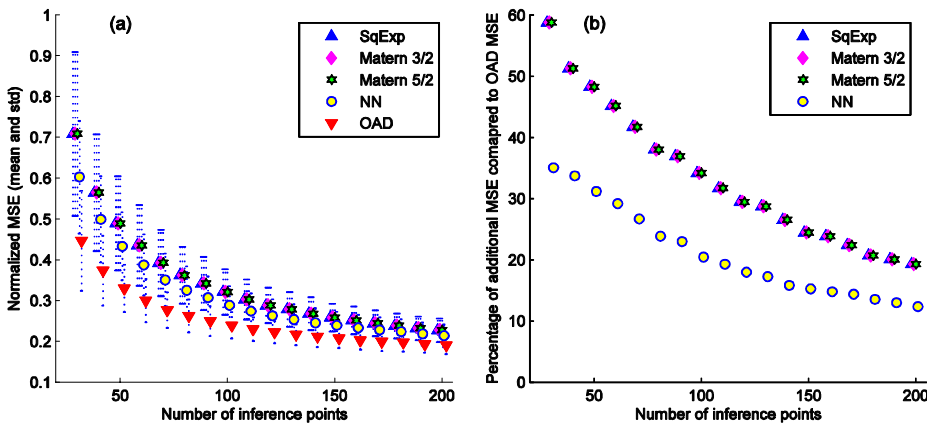


Fig. 2. (a) Normalized Mean Square Error (MSE) and (b) percentage of additional MSE compared to OAD MSE for the SIC dataset. The OAD covariance function systematically results in better estimates regardless of the chosen inference and testing sets.

¹ The SIC dataset can be downloaded at: <http://www.aigeostats.org>

6 Conclusions

This paper proposed a new non-stationary covariance function which allows simple geometric interpretation and depends on the angle at which points can be seen from an observation centre. Numerical evaluations with both artificial and real datasets demonstrate better predictive qualities for GP regression with the OAD covariance function than with other popular covariance functions. Although the main focus of this paper was on GPs, it is important to emphasize that the covariance function proposed is also a Mercer kernel and therefore can be applied to kernel machines such as support vector machines, kernel principal component analysis and others [6], [7]. The application of the derived covariance function to other kernel methods is an area of future work.

Acknowledgements

This work has been supported by the Rio Tinto Centre for Mine Automation and the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government.

References

1. Rasmussen, C.E., Williams, C.K.I.: *Gaussian Processes for Machine Learning*. MIT Press, Cambridge (2006)
2. Williams, C.K.I.: Computation with infinite neural networks. *Neural Computation* 10(5), 1203–1216 (1998)
3. Sugiyama, M., Hachiya, H., Towell, C., Vijayakumar, S.: Geodesic Gaussian kernels for value function approximation. In: *Proceedings of 2006 Workshop on Information-Based Induction Sciences*, Osaka, Japan, pp. 316–321 (2006)
4. Melkumyan, A., Ramos, F.: A Sparse Covariance Function for Exact Gaussian Process Inference in Large Datasets. In: *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence*, pp. 1936–1942 (2009)
5. Dubois, G., Malczewski, J., De Cort, M.: Mapping radioactivity in the environment. *Spatial Interpolation Comparison 1997* (Eds.). EUR 20667 EN, EC (2003)
6. Schölkopf, B., Smola, A.J.: *Learning with Kernels*. MIT Press, Cambridge (2002)
7. Bottou, L., Chapelle, O., DeCoste, D., Weston, J.: *Large-scale kernel machines*. The MIT Press, Cambridge (2007)

Appendix A: Detailed Derivation of the OAD Covariance Function

The covariance function is constructed by evaluating the integral in Eq. (6) where $h(\mathbf{x}, \mathbf{u}; \mathbf{x}_c)$ is defined in Eq. (5). The unit circle of integration $\|\mathbf{u} - \mathbf{x}_c\|_2 = 1$ is shown in Fig. 3 where diameters AB and $A'B'$ are introduced which are perpendicular to the vectors $\mathbf{x}_c \mathbf{x}$ and $\mathbf{x}_c \mathbf{x}'$, respectively. From Eq. (5) it follows that

$$h(\mathbf{x}, \mathbf{u}; \mathbf{x}_c) = \begin{cases} a_0 & \text{if } u \in \text{arc } AB'B \\ b_0 & \text{if } u \in \text{arc } BA'A \end{cases},$$

$$h(\mathbf{x}', \mathbf{u}; \mathbf{x}_c) = \begin{cases} a_0 & \text{if } u \in \text{arc } A'AB' \\ b_0 & \text{if } u \in \text{arc } B'BA' \end{cases},$$

so that the integrand $h(\mathbf{x}, \mathbf{u}; \mathbf{x}_c)h(\mathbf{x}', \mathbf{u}; \mathbf{x}_c)$ is equal to a_0^2 , a_0b_0 , b_0^2 and a_0b_0 on the arcs AB' , $B'B$, BA' and $A'A$, respectively. As the lengths of the arcs $B'B$ and $A'A$ are equal to $\alpha(\mathbf{x}, \mathbf{x}'; \mathbf{x}_c)$ and the lengths of the arcs AB' and BA' are equal to $\pi - \alpha(\mathbf{x}, \mathbf{x}'; \mathbf{x}_c)$, the integral in Eq. (6) can be calculated in closed form resulting in

$$k_2(\mathbf{x}, \mathbf{x}') = \sigma^2 \pi (a_0^2 + b_0^2) \left(1 - \frac{1 - 2a_0b_0 / (a_0^2 + b_0^2)}{\pi} \alpha(\mathbf{x}, \mathbf{x}'; \mathbf{x}_c) \right). \tag{17}$$

Defining φ from the equations $\cos(\varphi/2) = a_0 / \sqrt{a_0^2 + b_0^2}$, $\sin(\varphi/2) = b_0 / \sqrt{a_0^2 + b_0^2}$ and denoting $\sigma_0 = \sigma \left[\pi (a_0^2 + b_0^2) \right]^{1/2}$, the Eq. (17) becomes identical to Eq. (7).

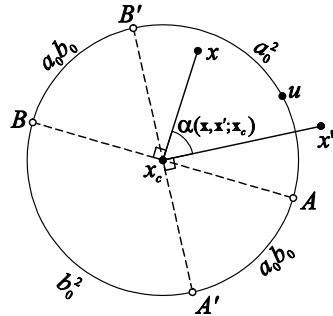


Fig. 3. Unit circle of integration

DOA Estimation of Multiple Convolutively Mixed Sources Based on Principle Component Analysis

Weidong Jiao^{1,2,*}, Shixi Yang², and Yongping Chang¹

¹ Dept. of Mechanical Engineering, Jiaying University, Jiaying, China 314001

² Dept. of Mechanical Engineering and Automation, ZheJiang University, Hangzhou, China 310027

jiaowd1970@mail.zjxu.edu.cn

Abstract. Direction of arrival (DOA) estimation is a basic task in array signal processing. A method based on principal component analysis (PCA) is presented for estimating DOA of multiple sources mixed convolutively. Convulsive mixtures of multiple sources in the spatio-temporal domain are firstly reduced to instantaneous mixtures by using the well-known short-time Fourier transformation (STFT) technique. From the time-frequency mixture in each frequency bin, one frequency response matrix of the mixing system from sources to sensors is estimated by the PCA based whitening. Furthermore, the DOAs of multiple sources are probed by using a whole estimating strategy. Consequently, all mixtures in total frequency bins contribute to a final estimation set, in which the source directions are shown as several direction clusters and/or local maxima. Experimental results indicate that the PCA based method has advantages over the well-known MUSIC (Multiple Signal Classification) method, especially under such conditions as the same number of sensors as sources, and closely placed sensors.

Keywords: MUSIC (Multiple Signal Classification), Principal Component Analysis (PCA), Short-Time Fourier Transformation (STFT), Direction of Arrival (DOA), Frequency Response Matrix, Whole Estimation Strategy.

1 Introduction

Direction of arrival (DOA) estimation is a basic and important task in array signal processing involved in many application fields such as wireless communication, audio/speech processing and radar signal processing [1]. Many source localization methods have been proposed. For example, the well-known MUSIC (Multiple Signal Classification) algorithm and its variants are popularly used for DOA estimation [2]. The MUSIC method is based on principle of subspace analysis. It identifies the noise subspace with second order statistics and search for location parameters that orthogonalize the steering vector and the noise subspace. However, the MUSIC method can only be applied under such hypothesis that there are fewer sources than sensors in an array. More unfortunately, its estimation performance for multiple source directions markedly deteriorates when sensors are closely placed.

* Corresponding author.

Principal component analysis (PCA) is an essential technique in data compression and feature extraction. It provides a way of reducing the number of input variables entering some data processing system so that a maximal amount of information is retained in the mean-square error sense; in addition, PCA provides uncorrelated components [3]. In this paper, we propose a PCA based frequency-domain method for DOA estimation of multiple sources mixed convolutively. In the proposed method, PCA is used for prewhitening mixture data in every frequency bin. Furthermore, the frequency response matrix from sources to sensors is approximated, followed by a whole estimation on all source directions at a clustering manner. Experimental results show that the PCA based method has advantage over the MUSIC based method, especially under such condition as the same number of sensors as sources and closely placed sensors.

2 Convolutive Mixing Modeling of Multiple Spatio-temporal Sources

Suppose that N source signals $s_j(t)$ are mixed and observed at a linear array with M sensors, i.e. $x_i(t) = \sum_{j=1}^N \sum_k h_{ij}(k) s_j(t-k)$, where $h_{ij}(k)$ represents the impulse response from source j to sensor i . Let d_i be the position of sensor i , and θ_j be the direction of source s_j (we suppose the direction orthogonal to the array is 90°).

Theoretically, the mixing process of multiple sources in a reverberant environment should be modeled as a convolutive mixture model [4]:

$$\mathbf{x}(t) = \mathbf{A}(t) * \mathbf{s}(t) + \mathbf{n}(t) \quad (1)$$

where $*$ denotes the convolution operation. $\mathbf{A} = \{A_{ij}, i, j = 1, 2\}$ is an unknown linear filter matrix, depending on transferring medium. $\mathbf{n}(t)$ is an additive noise vector.

We implement DOA estimation in the frequency domain. By L -point short time Fourier transformation (STFT), time-domain signals $x_i(t)$ are converted into frequency-domain time-series signals $X_i(f, m)$, where $f = 0, f_s/L, \dots, f_s(L-1)/L$ (f_s : sampling frequency), and m is the window frame index of filter. Thus, the convolutive mixtures in (1) are reduced to instantaneous mixtures in every frequency bin, i.e. $X_i(f, m) = \sum_{j=1}^N H_{ij}(f) s_j(f, m)$. Although in a reverberant condition, the frequency response $H_{ij}(f)$ can also be approximated as $H_{ij}(f) = e^{j2\pi f c^{-1} d_i \sin(\theta_j)}$, Considering the direction of source θ_j as spatio-directional variable θ , we have a steering vector

$$\mathbf{a}(f, \theta) = \left[e^{j2\pi f c^{-1} d_1 \sin(\theta)} \quad \dots \quad e^{j2\pi f c^{-1} d_M \sin(\theta)} \right]^T \quad (2)$$

where c is the propagation velocity. Then, the sensor observations can be modeled as [5]

$$\mathbf{X}(f, m) = \sum_{j=1}^N \mathbf{a}(f, \theta_j) s_j(f, m) = \mathbf{H}(f) \mathbf{s}(f, m) \quad (3)$$

where $\mathbf{X}(f, m)$ is a M -dimensional vector and $\mathbf{X}(f, m) = [X_1(f, m) \ \cdots \ X_M(f, m)]^T$. The present task is identifying the frequency response matrix $\mathbf{H}(f)$ (or the steering vector $\mathbf{a}(f, \theta_j)$, $j = 1, 2, \dots, N$) from the frequency-domain mixtures $\mathbf{X}(f, m)$ using some techniques such as PCA, and finally estimating the directions $\theta_1, \dots, \theta_N$ of all source signals.

3 The MUSIC Based DOA Estimation

In the well-known MUSIC algorithm [2], the correlation matrix $\mathbf{R} = \langle \mathbf{X}(f, m) \cdot \mathbf{X}(f, m)^H \rangle_m$ of sensor observations $\mathbf{X}(f, m)$ is calculated, where $(\cdot)^H$ represents a conjugate transpose and $\langle \cdot \rangle_m$ denotes the averaging operator. Then, implement the eigenvalue decomposition on the correlation matrix \mathbf{R} which producing $\mathbf{R} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^H$, $\mathbf{V} = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_M]$, $\mathbf{\Lambda} = \text{diag}[\lambda_1 \ \cdots \ \lambda_M]$, where \mathbf{v}_k is an eigenvector (M -dimensional column vector) and λ_k is the eigenvalue of \mathbf{v}_k sorted as $\lambda_1 \geq \cdots \geq \lambda_M$. The N points where the function $U(\theta) = \sum_{k=N+1}^M |\mathbf{v}_k^H \mathbf{a}(f, \theta)|^2$ approaches zero correspond to the directions $\theta_1, \dots, \theta_N$ of the source signals [5]. It is obvious that the MUSIC based method requires more sensors than sources in number.

4 The PCA Based DOA Estimation

4.1 Principal Component Analysis (PCA)

The basic goal in PCA is to reduce the dimension of the data. Indeed, it is well-known that the representation given by PCA is an optimal linear dimension reduction technique in the mean-square sense [6]. For an observed vector \mathbf{x} , the whitening means that the \mathbf{x} is linearly transformed to a vector $\mathbf{z} = \mathbf{Q} \mathbf{x}$ such that the covariance matrix of \mathbf{z} equals unity: $E\{\mathbf{z} \mathbf{z}^T\} = \mathbf{I}$. This transformation is always possible. For example, it can be accomplished by classical PCA [7]. In addition to whitening, PCA may allow us to determine the number of sources when there are more sensors than sources in an array (i.e. $M > N$). Its principle is that if noise level is low, the energy of \mathbf{x} is essentially

concentrated on the subspace spanned by the N first principal components, with N the number of sources in model (1).

According to the model (3), the time-frequency mixture in the k th frequency bin is written as

$$\mathbf{X}(f_k, m) = \sum_{j=1}^N \mathbf{a}(f_k, \theta_j) s_j(f_k, m) = \mathbf{H}(f_k) \mathbf{s}(f_k, m) \quad (4)$$

where $\mathbf{a}(f_k, \theta_j) = \left[e^{j2\pi f_k c^{-1} d_1 \sin(\theta_j)} \quad \dots \quad e^{j2\pi f_k c^{-1} d_M \sin(\theta_j)} \right]^T$ is the steering vector from the j th source to all M sensors. $\mathbf{X}(f_k, m)$ is a M -dimensional mixture vector in the k th frequency bin, and with

$$X_i(f_k, m) = \sum_{j=1}^N e^{j2\pi f_k c^{-1} d_i \sin(\theta_j)} s_j(f_k, m) \quad (5)$$

where $i = 1, \dots, M$, $j = 1, \dots, N$.

The PCA based whitening processing on the mixture $\mathbf{X}(f_k, m)$ gives

$$\mathbf{Z}(f_k, m) = \mathbf{Q}(f_k) \mathbf{X}(f_k, m) = \mathbf{Q}(f_k) \sum_{i=1}^M \sum_{j=1}^N e^{j2\pi f_k c^{-1} d_i \sin(\theta_j)} s_j(f_k, m) = \mathbf{B}(f_k) \mathbf{s}(f_k, m) \quad (6)$$

where $\mathbf{Q}(f_k)$ is a $M \times M$ whitening-transform matrix. Accordingly, $\mathbf{B}(f_k)$ can be looked as a $M \times N$ global-transform matrix.

4.2 Identification of the Frequency Response Matrix by the PCA Based Whitening

In DOA estimation of multiple spatio-temporal sources, the PCA based whitening is used not as a preprocessor to improve convergence performance of further data analysis (for example of independent component analysis [6]), but as an identifier to identify the frequency response matrix $\mathbf{H}(f_k)$ (or the steering vector $\mathbf{a}(f_k, \theta_j)$) from the k th bin mixture $\mathbf{X}(f_k, m)$.

It has been proved that the PCA based whitening transformation is orthogonal. That is, the global-transform $\mathbf{B}(f_k) = \mathbf{Q}(f_k) \mathbf{H}(f_k)$ in (9) is an orthogonal matrix. Furthermore, we have

$$\mathbf{Q}(f_k) = \mathbf{B}(f_k) \mathbf{H}(f_k)^\dagger \Rightarrow \mathbf{Q}(f_k)^{-1} = \mathbf{H}(f_k) \mathbf{C}(f_k) \quad (7)$$

where $(\cdot)^\dagger$ is pseudo inverse operator. Obviously, the matrix $\mathbf{C}(f_k) = \mathbf{B}(f_k)^\dagger$ is also orthogonal, which implies the frequency response $\mathbf{H}(f_k)$ can be orthogonally transformed into the inverse whitening matrix $\mathbf{Q}(f_k)^{-1}$. Under orthogonal transformations,

both *Euclidean* distance and direction of a vector are invariant. Also, the singular values of a matrix are invariant. [8]. Thus, we may say that the columns of $\mathbf{Q}(f_k)^{-1}$ span the same space as the columns of $\mathbf{H}(f_k)$, and the approximation $\hat{\mathbf{H}}(f_k) = \mathbf{Q}(f_k)^{-1}$ may be used for DOA estimation of multiple spatio-temporal sources. Alternatively, some other second-order methods such as factor analysis can also be used for identification the frequency response matrix, and be further used for DOA estimation of multiple sources.

4.3 DOA Estimation by Using a Whole Estimation Strategy

In PCA, there does exactly exist the inherent scaling and permutation ambiguities, just as that in ICA [9], which lead to the approximated $\hat{\mathbf{H}}(f)$ columns can have arbitrary scales and be permuted arbitrarily compared with the real frequency response of the mixing system. Furthermore, the element $\hat{H}_{ij}(f)$ of the matrix $\hat{\mathbf{H}}(f)$ may have an arbitrary amplitude. Inspired by the literature [5], the mixing system is also remodeled with attenuation A_{ij} (real-valued) and phase modulation $e^{j\varphi_j}$ at the origin, which leads to $\hat{H}_{ij}(f) = A_{ij} e^{j\varphi_j} e^{j2\pi f c^{-1} d_i \sin \theta_j}$. Therefore, the scaling ambiguity can be cancelled out by calculating the ratio between two elements $\hat{H}_{ij}(f)$ and $\hat{H}_{i'j}(f)$ corresponding to the same source j : $\hat{H}_{ij}/\hat{H}_{i'j} = A_{ij}/A_{i'j} e^{j2\pi f c^{-1} (d_i - d_{i'}) \sin \theta_j}$. Then, taking the angle yields a formula for estimating θ_j , i.e. $\hat{\theta}_j = \sin^{-1} \left(\text{angle} \left(\hat{H}_{ij} / \hat{H}_{i'j} \right) / 2\pi f c^{-1} (d_i - d_{i'}) \right)$. For the permutation ambiguities problem, a whole estimating strategy is proposed, its principle is described as follows:

(a) Given an whitening matrix $\mathbf{Q}(f)$ by PCA, approximate the frequency response of the mixing system by $\hat{\mathbf{H}}(f) = \mathbf{Q}^{-1}(f)$. The $\hat{\mathbf{H}}(f)$ is further written as: $\hat{\mathbf{H}}(f) = \{ \hat{H}_{ij}(f) \}$, where $\hat{H}_{ij}(f) = A_{ij} e^{j\varphi_j} e^{j2\pi f c^{-1} d_i \sin \hat{\theta}_j}$, $i = 1, 2, \dots, M$, $j = 1, 2, \dots, N$.

(b) Find all two-rows combinations of the $M \times N$ frequency response matrix $\hat{\mathbf{H}}(f)$. Theoretically, C_M^2 combinations can be obtained. For any combination, for example the combination $(p, q | p \neq q)$ formed by the p th and q th rows of the $\hat{\mathbf{H}}(f)$, an estimated DOA set can be computed by

$$\{ \hat{\theta}_j \}_{(p,q)} = \left[\sin^{-1} \left(\frac{\text{angle} \left(\hat{H}_{qj} / \hat{H}_{pj} \right)}{2\pi f c^{-1} (d_q - d_p)} \right) \right], j = 1, \dots, N \quad (8)$$

(c) Consequently, the C_M^2 combinations can give C_M^2 estimated DOA sets. Although $\hat{\theta}_j$ may not correspond to s_j but to another source signal because of the permutation ambiguity, all source directions can be obtained from the C_M^2 combinations. During the DOA estimation, $\hat{\theta}_j$ may becomes complex and only null direction is

obtained if the absolute value of the argument of \sin^{-1} is larger than 1. However, these null directions do not affect total DOA estimation of all sources under our whole estimation strategy.

The PCA based DOA estimation consists of three processing steps, i.e. (1) Signal domain transformation using the STFT, by which convolutive mixtures in the time domain are transformed into instantaneous mixtures in the frequency domain; (2) Frequency response estimation using the PCA based whitening, by which the frequency response matrix of a mixing system is approximated; (3) Whole DOA estimation and local peak detection, in which the source directions are wholly estimated using the formulation (14) and finally determined by local peak detection.

In order to evaluate the whole performance of a DOA estimation method, we use the *Euclidean* norm to define an error evaluator named Total Direction Mismatch (TDM), which is formulated as

$$TDM = \frac{\text{norm}\{[\hat{\theta}] - [\theta]\}}{\text{norm}\{[\theta]\}} = \sqrt{\sum_{j=1}^N (\hat{\theta}_j - \theta_j)^2} / \sqrt{\sum_{j=1}^N \theta_j^2} \quad (9)$$

where θ_j is the 'true' direction of the j th source and $\hat{\theta}_j$ is its estimate. $\text{norm}(\bullet)$ is the operator for solving *Euclidean* norm.

5 Experimental Results

We use the MUSIC method and the PCA based method for performing experiments to estimate the DOAs of three simulated source signals which are convolutively mixed. The initially experimental conditions are summarized in Table 1. We use two DOA estimators (i.e. the MUSIC [2] and the PCA based whitening [6]) to implement frequency response estimation of the convolutively mixing system.

The DOA estimation results under the initial conditions in the Table 1 are shown in figure 1.

Individual DOA estimation set by mixtures in each frequency bin is one by one described in the upper subplots whose x-coordinates and y-coordinates are all 'Frequency Bin (Hz)' and 'Direction (Deg)'. By the use of the whole estimation strategy

Table 1. Initially experimental conditions

Model of Source Signals	Coefficients of Sensor Array	Algorithms for Data Analysis
$\mathbf{s} = \begin{cases} s_1 = \sin(\omega_1 t) \\ s_2 = \sin(\omega_2 t) \cdot \sin(\omega_3 t) \\ s_3 \text{ is a white Gaussian noise} \end{cases} \begin{cases} \omega_1 = \pi/20 \\ \omega_2 = \pi/18 \\ \omega_3 = \pi/6 \end{cases}$	Array Type: Linear Number of Sensors: $M = 5$ Inter-element Spacing: $\Delta = 0.04$ (in wavelength) Angle Resolution: $\Delta\theta = 0.5^\circ$	Coefficients of the STFT: Length of Time Window: 32 Length of overlap: 16 Number of DFT: 256 DOA Estimators: MUSIC, PCA
Sampling Frequency and Length: $F_s = 2\text{Hz}$ Sampling Length: $T = 1024$ True DOA of Sources: $\theta_1 = 15^\circ, \theta_2 = 18^\circ, \theta_3 = 21^\circ$		

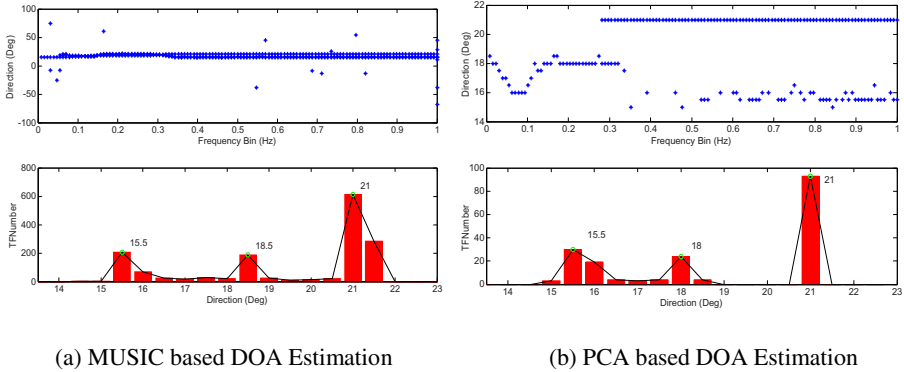


Fig. 1. DOA estimation result by the MUSIC and PCA based DOA estimators

combined with the local peak detection, we finally estimate and show all sources directions in the bottom subplots whose x-coordinates and y-coordinates are all 'Direction (Deg)' and 'TFNumber', i.e. abbreviation of 'Total Frequency Number', which enables one to evaluate total contribution of all DOA estimation sets synthetically. It can be seen that the two DOA estimators (i.e. the MUSIC and the PCA based whitening) both perform well, under the initially experimental condition shown in Table 1. Two DOA estimation groups is $[15.5^\circ, 18^\circ, 21^\circ]$ and $[15.5^\circ, 18.5^\circ, 21^\circ]$, and their total direction mismatch TDM is 0.0159 and 0.0225 separately.

To further compare the two DOA estimators, we design some controlled conditions for experiment which are summarized in Table 2. Under the controlled conditions, total performances of the two DOA estimators, i.e. the MUSIC and the PCA based whitening, are comparatively analyzed and shown in figure 2.

In the figure 2, the two DOA estimators are, under different controlled conditions, compared to each other in different row-plots. In the upper row-plot (a), DOA estimations under different *Number of Sensors* are shown. It can be seen that, when the number of sensors is changed to 3 (equal to the number of sources), remarkable estimation error ($TDM = 0.4767$) is brought by the MUSIC estimator (see the red solid-square line '-□-'), because of unsuccessful estimation of the first source direction

Table 2. Controlled conditions for further experiment

Changed Coefficients	Valued Range of the Changed coefficient
Number of Sensors: M	3, 4, 5, 6, 7
Angle Resolution: $\Delta\theta$	0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 1.0, 1.5, 2.0 Degree
Inter-element Spacing: Δ	0.00005, 0.00025, 0.0025, 0.01, 0.02, 0.04 (in wavelength)
Note: When one controlled coefficients is used for performance evaluation of one DOA estimators, the other coefficients keep unchanged just as that shown in TABLE 1.	

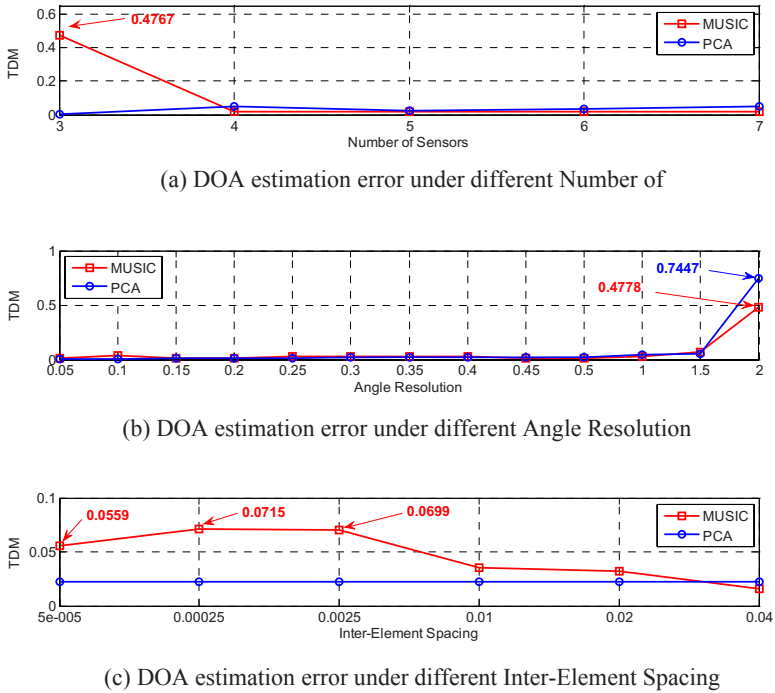


Fig. 2. Total performance of the two DOA estimators under the controlled conditions in Table 2

$\theta_1 = 15^\circ$. Contrastively, all three source directions are successfully captured by the PCA based method (see the blue solid-circle line '-o-'). When the number of sensors is larger than the number of sources (varying from 4 to 7), the two DOA estimators both work well and all source directions are estimated effectively.

The second changed coefficient *Angle Resolution* has similar influence upon the two estimators, which is shown in the middle row-plot (b). When the angle resolution increases up to $\Delta\theta = 2^\circ$, the MUSIC estimator cannot estimate the first source direction, and its error reaches to 0.4778. For the PCA based estimator, two source directions are mistakenly estimated and its *TDM* error reaches to 0.7447. As the angle resolution decreases, good estimation result is obtained by whichever estimator, which implies the importance to select an appropriate angle resolution.

Finally, we change the third controlled coefficient *Inter-Element Spacing* from 0.00005 to 0.040 (in wavelength). The DOA estimation results are described in the bottom row-plot (c). It can be clearly seen that the *Inter-Element Spacing* coefficient has strong impact on performance of the MUSIC based estimator. When too small inter-element spacing (for example $\Delta = 0.00005$, 0.00025 or 0.0025) is chosen, remarkable DOA estimation errors (i.e. $TDM = 0.0559$, 0.0715 and 0.0699) are brought by the MUSIC estimator. Whereas, the PCA based estimation is not interfered by different inter-element spacing at all, and all source directions are correctly estimated.

6 Conclusions

The experimental results show the PCA based method has some advantages over the MUSIC method. Theoretically, the MUSIC method requires more sensors than sources in number. When such requirement is not satisfied, its performance for DOA estimation will deteriorate markedly. For the PCA based estimator, the above requirement can be loosened to some extent. That is, the number of sensors is not less than the number of sources in an array. Furthermore, the PCA based estimator can implement precise direction estimation of multiple sources mixed convolutively, even if the sensors are closely placed (i.e. with small inter-element spacing). All of these imply its wider potential applicability than the MUSIC based method.

In the experiments, we notice that accurate DOA estimation of all sources is still accomplished within certain a controlled angle resolution, even if the whitening matrix by PCA is close to singular or badly scaled, which indicates strong numeric robustness of the propose method. In addition, it is very important to choose an appropriate angle resolution for all DOA estimators, because it does directly affect total performance of a DOA estimator. In practice, this coefficient should be carefully chosen according to different application purposes.

Acknowledgement

This work is partially supported by the 863 Plan of China Grant # 2007AA04Z424, National Science Foundation of China Grant # 50505016 and Science and Technology Plan of Zhejiang Province, China Grant # 2007C21041.

References

- [1] Brandstein, M., Ward, D. (eds.): *Microphone Arrays*. Springer, Heidelberg (2001)
- [2] Schmidt, R.O.: Multiple emitter location and signal parameter estimation. *IEEE Trans. Antennas and Propagation* 34, 276–280 (1986)
- [3] Oja, E., Karhunen, J., Hyvarinen, A.: From Neural PCA to Neural ICA. In: *NIPS Post-Conference Workshop on Blind Signal Processing, Snowmass, Colorado*, pp. 1–13 (1996)
- [4] Lucas, P., Clay, S.: Convolutional blind separation of non-stationary sources. *IEEE Trans. on Speech and Audio Processing* 8(3), 320–327 (2000)
- [5] Sawada, H., Mukai, R., Makino, S.: Direction of arrival estimation for multiple source signals using independent component analysis. In: *Proc. International Symposium on Signal Processing and its Applications*, pp. 411–414 (2003)
- [6] Hyvarinen, A.: Survey on independent component analysis. *Neural Computing Surveys* 2, 94–128 (2001)
- [7] Oja, E.: The nonlinear PCA learning rule in independent component analysis. *Neurocomputing* 17(1), 25–46 (1997)
- [8] Schott, J.R.: *Matrix analysis for statistics*. Wiley-Interscience, Hoboken (2005)
- [9] Comon, P.: Independent component analysis, a new concept? *Signal Processing* 36, 287–314 (1994)

Weighted Data Normalization Based on Eigenvalues for Artificial Neural Network Classification

Qingjiu Zhang and Shiliang Sun

Department of Computer Science and Technology, East China Normal University,
500 Dongchuan Road, Shanghai 200241, P.R. China
qjzh08@gmail.com, slsun@cs.ecnu.edu.cn

Abstract. Artificial neural network (ANN) is a very useful tool in solving learning problems. Boosting the performances of ANN can be mainly concluded from two aspects: optimizing the architecture of ANN and normalizing the raw data for ANN. In this paper, a novel method which improves the effects of ANN by preprocessing the raw data is proposed. It totally leverages the fact that different features should play different roles. The raw data set is firstly preprocessed by principle component analysis (PCA), and then its principle components are weighted by their corresponding eigenvalues. Several aspects of analysis are carried out to analyze its theory and the applicable occasions. Three classification problems are launched by an active learning algorithm to verify the proposed method. From the empirical results, conclusion comes to the fact that the proposed method can significantly improve the performance of ANN.

Keywords: Artificial neural network, Principle component analysis, Weighted data normalization, Active learning.

1 Introduction

Artificial neural network (ANN), usually called “neural network” (NN), is an information processing paradigm inspired by biological nervous systems. The mathematical model of ANN is constructed by lots of “neurons” which often work together, usually in the form of hierarchy, to solve learning problems. Each neuron has a threshold function which can be continuous or discrete. ANN often has several layers, the former layer’s outputs are weighted and used as the inputs of the next layer. The function of the latter layer maps the inputs to its outputs which will be weighted again and used as the inputs of the next layer. Therefore, when the threshold functions are selected, all the efforts are to find the weights.

ANN has wide applications, and it also has its shortcomings. Almost all the learning problems, such as classification, progression and multitask learning [1] can be solved by ANN. Relative researches have show that ANN can solve not only linear problems, but also nonlinear problems. Moreover, it has been proven that ANN with three layers can fit any non-linear problems [2]. When an neuron of the neural network fails, others can function without any problem by

their parallel characteristics. Although ANN has lots of advantages, it has some defects. Many kinds of ANNs are prone to step into local minimum problems, which means a single ANN is not a stable learner. Ensemble techniques sometimes can overcome this kind of problem by combining several ANNs [3,4,5]. In addition, the settings of ANN's parameters, such as the number of hidden neurons and the learning rate, are strongly depended on the characters of the data set. Therefore, there is none fixed rule to set them. Usually, expert knowledge plays an important role in setting those parameters for concrete issues.

The improvements for neural network can be mainly divided into two aspects: changing ANN's architecture and normalizing the original data. Changing the architecture of neural network sometimes means constructing new types of ANNs. Researches on ANN may start from the single-layer neural network. Single-layer network has simple input-output relations, thus sheds light on the research of multi-layer network [6]. Based on different theories and thoughts, different types of ANNs have been put forward, such as feedforward neural network, radial basis function (RBF) network, hopfield network and so on. They are based on different mathematical models which are suitable to solve different learning problems.

Normalizing the data is another manner to improve the accuracy of ANN. In many neural network applications, raw data (not preprocessed or not normalized) is used. However, raw data suffer lots of problems including high dimensional and time-consuming problems. By normalizing the data, ANN can get better effects and save much time for training. Using correlation coefficients as weights for input variables can significantly boost ANN [7]. Song and Kasabov also presented their preprocessing data method WDN-RBF for radial basic function typed neural networks [8]. Furlanello and Giuliani have normalized raw data by combining local and global space transformations [9].

This paper focuses on the data normalization approach for ANN. A new data normalization method WDNE (Weighted Data Normalization based on Eigenvalues) which weights the data by eigenvalues is proposed. WDNE is different from existing data normalization methods. It leverages the fact that the features which have different potentials in learning problems should play different roles. The data set is weighed by the eigenvalues, which means some features are enhanced while others are weakened. WDNE firstly uses principle component analysis (PCA) to rebuild the data set to ensure the features are uncorrelated. Then all the features are weighted by their corresponding eigenvalues.

The rest of the paper is organized as follows. Section 2 describes our proposed normalization process and gives the corresponding analysis. Section 3 reports experimental results. At last, conclusions are drawn in Section 4.

2 WDNE and Analysis

WDNE is based on PCA. It can be regarded as a method that induces the weights in ANN to change, and it can also be regarded as an approach which preprocesses the data before applied.

2.1 PCA

PCA is mathematically defined as an orthogonal linear transformation that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. PCA can be used to reduce the high dimensionality of dimensional data and improve the predictive performance of some machine learning methods [10]. High dimensional data often cause computational problems and run the risk of overfitting. Furthermore, many redundant or highly correlated features may probably cause a degradation of prediction accuracy. By simply discarding some features which have little information, PCA can eliminate the problem of high dimensionality.

The process of PCA aims to transform a problem from its natural space into another space, in which all the mapped features have irrelevant relationships. Suppose that $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ is the data set which contains n examples, and that the problem has d dimensions, which means every example has d random variables. For example, a example can be expressed as $\mathbf{x}_i = \{x_i^1, x_i^2, \dots, x_i^d\}$. When the data set \mathbf{X} is available, covariance matrix can be calculated in the form of $\Sigma = E\{(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T\}$, where E means figuring out the mathematical expectation, and $\boldsymbol{\mu}$ is the mean of the examples. Σ is usually figured out by the way of $\Sigma = 1/n \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$. By mathematical calculating, the eigenvalues $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, \dots, \lambda_d\}$ and the eigenvectors $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_d\}$ of the covariance matrix Σ can be obtained. Subsequently, the initial example \mathbf{x} is mapped to a new space in the form of $\mathbf{x}' = \mathbf{U}^T \mathbf{x}$. In the new space, every feature of the example is uncorrelated. Unless stated otherwise, the k th principle component will be taken to mean the principle components with the k th largest eigenvalue [11]. By selectively choosing some principle components, the high dimensional space is transformed into a fitting subspace.

2.2 Weighted Data Normalization Based on Eigenvalues

The basic idea of the proposed method is that the more useful a component is the more important role it should play. The data are weighted by the eigenvalue vector $\boldsymbol{\lambda}$. Every component x_i^j of the input \mathbf{x}_i which is processed by PCA has a corresponding eigenvalue λ_j . λ_j has lots of potential contents, one of which is that it indicates the variance of the component. The one which has the bigger eigenvalue can clearly provide more information, and PCA mainly depends this point for dimensionality reduction. However, if the data set is just processed by PCA, all the selected components still have equal roles in the training process. Actually, the component providing more information should play more decisive role in solving learning problems. Therefore, all the components can be weighted by their corresponding eigenvalues.

WDNE can be divided into two steps. Initially, it processes the data by PCA. Then, the processed data are weighted by the eigenvalues which are figured out in the first step. To avoid computational problems, the eigenvalues are normalized before weighting the data. The eigenvalues are normalized in the form of $\lambda'_j = \lambda_j / \lambda_1$, where λ_1 is the biggest eigenvalue. Subsequently, each feature is weighted

by the corresponding eigenvalue in the form of $x_i^j = x_i^j \lambda_j'$. If the original data set has high dimensionality, it can be reduced by discarding some features which provide little information.

2.3 Analysis

WDNE reinforces the principle components with large eigenvalues and weakens the others. If the PCA-processed data are directly used, all the components will play an equal role in the training process. However, if the data are weighted by WDNE, the principle components will play more important roles than the sub-principle components. It can be apparently analyzed from the data processing. The principle component multiplies a relative large constant, which means the distance between elements will be enlarged. When applied into learning problems, the principle components will play a more important role in deciding the final hypotheses. Similarly, the components which provide little information are weakened, because they can not provide good decisions. From this point of view, the PCA, in which the selected components are weighted by one while the unselected components are weighted by zero, can be seen as a special situation of WDNE.

WDNE can also be regarded as a method which enhances the effects of ANN by weighting some of the inputs. Initially, the output $net = \mathbf{w}^T \mathbf{x} = \sum_{j=1}^d w_j x_j$ of the input layer is mapped to the function f of the hidden layer. If there is no WDNE, the output of the hidden layer is $y = f(net)$. When WDNE is used, the output of the hidden layer will be $y = f(net') = f(\mathbf{w}^T \mathbf{\Lambda} \mathbf{x})$, where $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues as its values. WDNE apparently does not change the architecture of ANN, instead, it improves the performance by weighting the inputs. The process can be explicitly described in Fig. 1.

WDNE is probably sensible to noise. Suppose some noisy features are contained into the learning problem. The noisy features would be transformed into a new space after PCA processing. If many of the noisy ingredients are

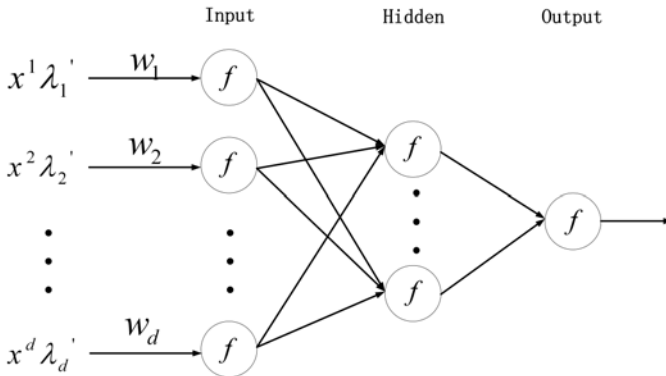


Fig. 1. WDNE

redistributed into the last several features, the noisy effect is obviously reduced. However, if many of the noisy ingredients are transformed into the first few features which correspond to the biggest eigenvalues, the noisy effect will be enhanced, and the performance of the recognition will be degraded. Consequently, caution should be taken before WDNE is applied to the problem which may contain noises.

3 Experiments

Back propagation neural network is applied in the implemented experiments. All of the data sets come from the UCI repository¹. The ANN has three layers, and the number of hidden neurons is set as

$$\text{neuron_number} = \sqrt{n + m} + a , \quad (1)$$

where n and m denote the number of input and output neurons of the networks respectively, and a is a constant ranging from one to ten. In order to effectively boost WDNE, five ANNs are used in every experiment, and the final hypothesis is the voted result by the five committees.

Active learning algorithm and ten-fold cross validation (CV) are lunched on the three experiments. Active learning aims to reduce the number of labeled data for learning by selecting the most informative examples [12]. The active learning algorithm used in this paper is for classification. Initially, only a few labeled examples are prepared at hand. At every round of iteration α examples are selected from β candidates which are the most valuable ones in the unlabeled examples pool. In order to more objectively reflects the result, ten-fold CV is applied. The data set is divided in to ten parts. One part is used as validation set, another part is used as test data, and the others are used for training. The final result is the average of the ten results.

3.1 Comparative Methods

In order to clearly evaluate the advantage of WDNE, three other representations of data are implemented, and they are original data (raw data), PCA processed data and negatively weighted data. The descriptions of them are listed as follows:

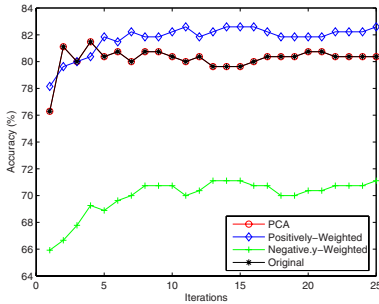
- **Original data:** This data set is the original (raw) data set.
- **PCA processed data:** This data set is just processed by PCA.
- **Positively weighted data:** This data set is processed by WDNE. In other words, this data set is firstly processed by PCA, then all the components is weighted by the corresponding eigenvalues.
- **Negatively weighted data:** This data set is firstly processed by PCA, then each component is weighted by the reversed eigenvalue $\lambda'_j = \lambda_{d-j+1}/\lambda_1$.

All the compared data are trained by the back-propagation network under the same setting of parameters.

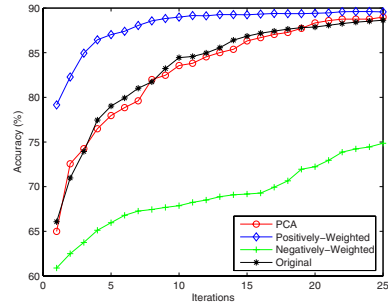
¹ <http://archive.ics.uci.edu/ml/>

3.2 Heart Classification

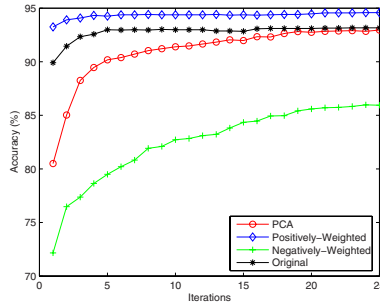
This is a two-class classification problem and the data set contains 150 positive examples and 120 negative examples. The initial dimension space consists of 13 features. The labeled data used for active learning algorithm initially contains six positive examples and six negative examples. The size of the pool is 100. At every round of iteration, four unlabeled examples are randomly selected from 16 candidates which are regarded as the most valuable ones. The final results can be seen from Fig. 2 (a).



(a) Heart classification



(b) Spam classification



(c) Waveform classification

Fig. 2. Classification performances

In Fig. 2 (a), the effect of the original data is almost equal to that of the data which are just processed by PCA. Negatively weighted data set has the worst performance, while positively weighted data set is clearly the best representation.

3.3 Spam Classification

This data set is composed of 4501 examples which contains 1813 spams. The problem have totally 57 features without dimension reduction. Initially, the active learning algorithm runs on 15 labeled examples which consists six positive

ones and nine negative ones. Before sampling, 32 unlabeled example which are seeded as the most informative ones are chosen from the pool whose size is 240. Subsequently, 10 examples are chosen from the 32 candidates at random.

Fig. 2 (b) shows that WDNE is significantly superior to the other three kinds of representations. As the results of last experiment, the data set of negative weighted has the worst performance. Although there is difference between the performances of original data and PCA processed data, they almost have the same effect on the whole.

3.4 Waveform Classification

This data set is also a binary classification which contains 1653 positive examples and 1655 negative examples. There are 40 features. It is worth noting that 19 of the 40 features are noises. Six positive examples and six negative examples are labeled at first. At each round of iteration, the algorithm firstly chooses 32 most informative examples from a pool whose size is 200. Subsequently, eight examples are randomly selected from the 32 candidates.

Fig. 2 (c) shows the results. It is apparent to tell that the WDNE performs the best and the negatively weighted data set plays the worst. In this experiment, it is obviously that PCA processed data are not as good as the original data. There may be several possible causes of this phenomenon. (1) It is caused by the noises. (2) The original representation of the data set is more suitable to solve the learning problem in this experiment.

4 Conclusions

In this paper, a new data normalization approach WDNE which can be used to improve the performances of neural networks is proposed. WDNE does not optimize the architecture of the ANN. However, it boosts the performance of ANN by preprocessing the data. PCA plays an important role in this method. The components are weighted by the corresponding eigenvalues of the covariance matrix. In order to verify the proposed method, three other representations of data are implemented in the experiments. All the utilized data sets come from the UCI repository. The empirical results clearly show that WDNE is an effective method for optimizing the performance of ANN.

Researches on WDNE requires further investigation. In this paper WDNE is used for ANN, it may be applied to combine with other approaches, such as distance metric learning (DML) [13], support vector machines (SVM) [14], etc. Moreover, WDNE may be regraded as a way for feature selection.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Project 60703005, and by Shanghai Educational Development Foundation under Project 2007CG30.

References

1. Jin, F., Sun, S.: Neural Network Multitask Learning for Traffic Flow Forecasting. In: Proceedings of the International Joint Conference on Neural Networks, pp. 1898–1902 (2008)
2. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley and Sons, New York (2001)
3. Breiman, L.: Bagging Predictors. *Machine Learning* 24(2), 123–140 (1996)
4. Freund, Y., Schapire, R.E.: A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence* 14(5), 771–780 (1999)
5. Sun, S.: Ensemble Learning Methods for Classifying EEG Signals. In: Haindl, M., Kittler, J., Roli, F. (eds.) MCS 2007. LNCS, vol. 4472, pp. 113–120. Springer, Heidelberg (2007)
6. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
7. Anderson, H., Black, T.: Multivariate Data Analysis. Prentice-Hall, London (1998)
8. Song, Q., Kasabov, N.: WDN-RBF: Weighted Data Normalization for Radial Basic Function Type Neural Networks. In: Proceedings of IEEE International Joint Conference on Neural Networks, pp. 2095–2098 (2004)
9. Furlanello, C., Giuliani, D.: Combining Local PCA and Radial Basis Function Networks for Speaker Normalization. In: Proceedings of IEEE Workshop on Neural Networks for Signal Processing, pp. 233–242 (1995)
10. Howley, T., Madden, M.G., O’Connell, M., Ryder, A.G.: The Effect of Principle Component Analysis on Machine Learning Accuracy with High Dimensional Spectral Data. In: Proceedings the 25th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, pp. 209–222 (2005)
11. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer, New York (2002)
12. Tong, S.: Active Learning: Theory and Applications. PhD thesis, Stanford University, Stanford (2001)
13. Yang, L.: Distance Metric Learning: A Comprehensive Survey. Michigan State University, Michigan (2006)
14. Cristianini, N., Taylor, J.S.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)

The Optimization of Kernel CMAC Based on BYY Learning

Guoqing Liu¹, Suiping Zhou¹, and Daming Shi²

¹ School of Computer Engineering, Nanyang Technological University, 639798, Singapore

² School of Electrical Engineering and Computer Science, Kyungpook National University, Daegu 702701, South Korea

Liug0008@ntu.edu.sg, Asspzhou@ntu.edu.sg, Dmshi@ee.knu.ac.kr

Abstract. Cerebellar Model Articulation Controller (CMAC) has attractive properties of fast learning and simple computation. The kernel CMAC, which provides an interpretation for the classic CMAC from the kernel viewpoint, strengthens the modeling capability without increasing its complexity. However, the kernel CMAC suffers from the problem of selecting its hyperparameter. In this paper, the Bayesian Ying-Yang (BYY) learning theory is incorporated into kernel CMAC, referred to as KCMAC-BYY, to optimize the hyperparameter. The BYY learning is motivated from the well-known Chinese Taoism Yin-Yang philosophy, and has been developed in this past decade as a unified statistical framework for parameter learning, regularization, structural scale selection and architecture design. The proposed KCMAC-BYY achieves the systematic tuning of the hyperparameter, further improving the performance in modeling capability and stability. The experimental results show that the proposed KCMAC-BYY outperforms the existing representative techniques in the research literature.

Keywords: Bayesian Ying-Yang learning theory, CMAC, kernel machine.

1 Introduction

The Cerebellar Model Articulation Controller (CMAC) [1] is a type of neural network based on a model of the mammalian cerebellum. Originally the CMAC was proposed as a function modeler for robotic controllers in 1975, but it has been extensively used in reinforcement learning and also as a classifier. As an associative memory neural network model, the CMAC has some attractive features of fast learning, simple computation, local generalization, and the fact that it can be realized by specialized high-speed hardware [2]. In addition, the application of the CMAC can be found in many areas such as robotic control, signal processing, and pattern recognition. Fig. 1 shows the architecture of CMAC model.

Practically, a huge problem always constrains the application of CMAC: the memory requirement grows exponentially with respect to the input dimension. In order to reduce the complexity of the CMAC, Albus introduced hash coding into his model [1]. This approach effectively reduces the size of memory, but it can result in collisions of the mapped weights and bring certain adverse impacts to the convergence of learning [3]. Another method, of decomposing a multivariate case into a group of

lower dimensional ones, is also widely used to reduce complexity in CMAC research. Although all the above architectures are less complex compared to the classic CMAC, they are more time consuming in the training process [3].

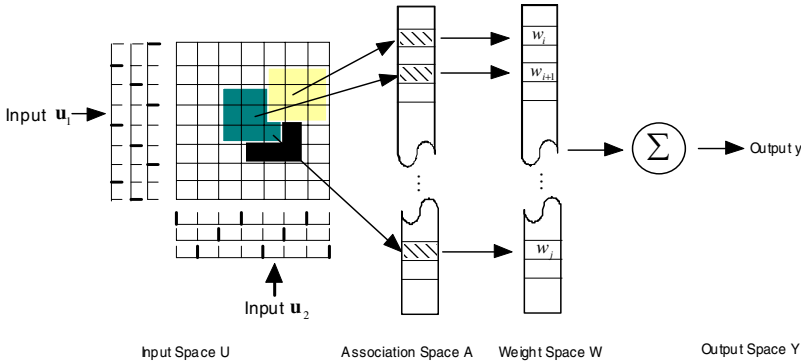


Fig. 1. Architecture of the Classic CMAC

Interpreting the CMAC as a type of kernel machine, the Kernel Cerebellar Model Articulation Controller (KCMAC) can reduce the complexity of the CMAC, and strengthens its modeling capability remarkably [3]. In order to further improve the modeling capability of KCMAC, one hyperparameter is introduced to penalize the mis-regression. Such a hyperparameter must be optimized to suit a specific problem; however, it is always pre-defined based on empirical knowledge in the original KCMAC. To address this problem, in this paper we attempt to tune the hyperparameter systematically using the Bayesian Ying-Yang learning theory.

Bayesian Ying-Yang (BYY) learning [4], [5], [6], is a statistical learning theory for a two pathway intelligent system via two complementary Bayesian representations of the joint distribution on the external observation and its inner representation, with all unknowns in the system determined by a principle that two Bayesian representations become best harmony. In our previous work, BYY has been successfully applied to the fuzzification phase of the CMAC [11], which is further incorporated with an online expectation-maximization (EM) algorithm to process time series data [13]. In this research, a novel KCMAC-BYY is proposed to achieve the systematic tuning the hyperparameter, and further improves the performance in modeling capability and stability.

The remainder of this paper is organized as follows. The KCMAC is briefly described in the next section. Section 3 introduces the BYY supervised learning into the KCMAC. The hyperparameter optimization using BYY learning is proposed in Section 4. Experiments and the detailed analyses are presented in Section 5, followed by the conclusions and future work in Section 6.

2 Overview of Kernel CMAC

To improve modeling capability of CMAC without increasing model complexity, a kernel version of CMAC is introduced by G. Horváth. In KCMAC, the association space is treated as the feature space of a kernel machine [3]. Considering the fact that

the binary basis functions can be regarded as first-order B-spline functions of fixed positions, the higher-order B-spline kernel functions can be designed artificially to replace the binary basis functions of the CMAC.

The weight vector \mathbf{w} in the KCMAC is determined by the following constrained optimization using a quadratic loss function [10]:

$$\text{Min}_{\mathbf{w}, \mathbf{e}} J_1(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{\gamma}{2} \sum_{i=1}^n e_i^2 \tag{1}$$

such that

$$z_i = \mathbf{w}^T \phi(\mathbf{u}_i) + e_i \tag{2}$$

where γ is the mis-regression penalty parameter of model, e_i is the error of i th input point, n is the number of data points, and $\phi(\mathbf{u}_i)$ corresponds to the mapping function in kernel machine to replace the ‘‘AND’’ operation of the classic. In addition to the classic binary CMAC, this kernel interpretation can also be used in higher order CMACs [7], [8], [9], with higher order basis functions. A CMAC with k th-order B-spline basis function corresponds to a kernel machine with $2k$ th-order B-spline kernels.

Then introducing Lagrangian, the response of the network will be

$$y(\mathbf{u}) = \phi(\mathbf{u}) \sum_{i=1}^n \alpha_i \phi(\mathbf{u}_i) = \sum_{i=1}^n \alpha_i K(\mathbf{u}, \mathbf{u}_i) \tag{3}$$

where α_i are the lagrange multipliers. In addition, by adding a regularization term into (1) and (2) respectively, the KCMAC can be easily extended to a regularized version, which has a better generalization capability.

In the KCMAC, the modeling capability can be reinforced greatly without increasing its complexity, but the hyperparameter γ , which is the mis-regression penalty parameter of the model, is introduced into the model. In the original KCMAC, this hyperparameter is always determined by empirical knowledge, while different values of γ will result in quite different performances. To guarantee the modeling capability and stability, the BYY learning is embedded into the KCMAC in this paper, since it can optimize γ systematically.

3 The Kernel CMAC with BYY Learning

In the BYY supervised learning [4], [5], [6], there are three primary elements: the inner representation \mathbf{w} , the external observations \mathbf{u} , and the output action z . The \mathbf{u} and z are known (visible), but the \mathbf{w} is unknown (invisible). All of these elements are treated as random variants, and the joint distribution $p(\mathbf{u}, z, \mathbf{w})$ can be calculated in two ways:

$$\begin{cases} p_{ying}(\mathbf{u}, z, \mathbf{w}) = p(\mathbf{w})p(\mathbf{u}|\mathbf{w})p(z|\mathbf{w}, \mathbf{u}) \\ p_{yang}(\mathbf{u}, z, \mathbf{w}) = p(\mathbf{u})p(z|\mathbf{u})p(\mathbf{w}|\mathbf{u}, z) \end{cases} \tag{4}$$

Practically, the results of these two equations are always not equal unless \mathbf{w} is the optimal solution. Notice that \mathbf{u} and \mathbf{w} are dialectical: in training \mathbf{u} and z are known but \mathbf{w} is unknown, and \mathbf{w} is obtained in terms of \mathbf{u} and z , while in testing or ning, \mathbf{w} is known but \mathbf{u} and z are unknown, and \mathbf{w} decides what \mathbf{u} and z are. The core idea of the BYY learning is that the specification of a Ying-Yang pair above enhances best the so-called Ying-Yang harmony.

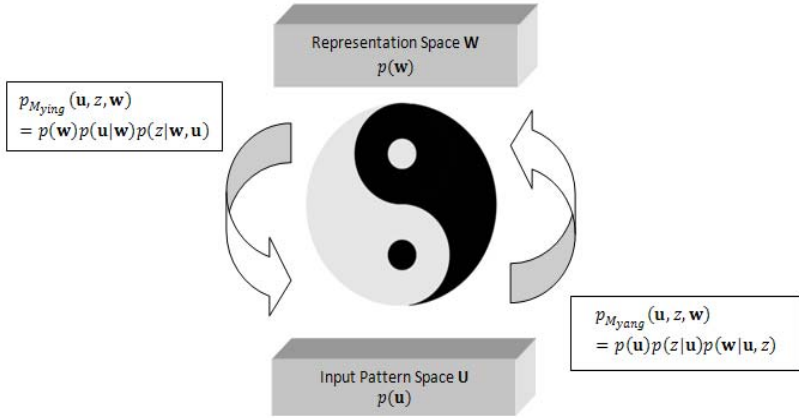


Fig. 2. BYY learning in the KCMAC

The KCMAC can be considered a system to obtain an optimal weight vector in terms of the training data. It is reasonable that in our work we design input data as the external observation \mathbf{u} , the desired output as the output action z , and the weight vector as the inner representation \mathbf{w} , while the optimal value of hyperparameter γ will make the model to be the best harmony. Fig. 2 depicts the BYY learning in the KCMAC.

4 Hyperparameter Optimization Using BYY Learning

4.1 Architecture Design for the KCMAC-BYY

The architecture design will be made by specifying the architecture of nents, $p(\mathbf{u})$, $p(z|\mathbf{u})$, $p(\mathbf{w}|\mathbf{u}, z)$, $p(\mathbf{w})$, $p(\mathbf{u}|\mathbf{w})$, $p(z|\mathbf{w}, \mathbf{u})$ in the Ying-Yang pair of the KCMAC-BYY.

- $p(\mathbf{u})$ and $p(z|\mathbf{u})$ with a given training data, are usually fixed on the kernel estimate [12] respectively.
- $p(\mathbf{w}|\mathbf{u}, z)$, called the coordination terminal in the BYY supervised learning, makes the invisible representation space W coordinate with the two visible spaces U, Z . In this paper, we set $p(\mathbf{w}|\mathbf{u}, z)$ to be free.

To specify $p(\mathbf{u}|\mathbf{w})$, $p(z|\mathbf{w}, \mathbf{u})$ and $p(\mathbf{w})$, we return back to the constrained optimization of the KCMAC in (1). Here we rewrite Eq. (1) as follows:

$$\text{Min}_{\mathbf{w}} J_1(\mathbf{w}) = \frac{\mu}{2} \mathbf{w}^T \mathbf{w} + \frac{\xi}{2} \sum_{i=1}^n e_i^2 \tag{5}$$

where γ is replaced by two new hyperparameters, μ and ξ , correspondings to $\gamma = \xi/\mu$. To interpret the KCMAC probabilistically, one can regards the Eq. (5) as defining a negative log-posterior probability for the weight vector \mathbf{w} , given a training data \mathcal{D} :

$$p(\mathbf{w}|\mathcal{D}, \ln \mu, \ln \xi) \propto \exp\left(-\frac{\mu}{2} \mathbf{w}^T \mathbf{w} - \frac{\xi}{2} \sum_{i=1}^n e_i^2\right) \tag{6}$$

where $p(\mathbf{w}|\mathcal{D}, \ln \mu, \ln \xi) \propto p(\mathbf{w}|\ln \mu)p(\mathcal{D}|\mathbf{w}, \ln \xi)$. Then we have

- $$p(\mathbf{w}|\ln \mu) = \left(\frac{\mu}{2\pi}\right)^{\frac{M}{2}} \exp\left(-\frac{\mu}{2} \mathbf{w}^T \mathbf{w}\right) \tag{7}$$

where M is the dimension of the weight vector or feature space.

- $$p(\mathbf{u}|\mathbf{w})p(z|\mathbf{w}, \mathbf{u}) = p(\mathbf{u}, z|\mathbf{w}, \ln \xi) = \sqrt{\frac{\xi}{2\pi}} \exp\left(-\frac{1}{2} \xi e^2\right) \tag{8}$$

where ξ is the mis-regression penalty parameter of the model, and e is the error of the input point.

4.2 Separation Functional of the KCMAC-BYY

In this paper, the Kullback-Leibler (KL) divergence [14] is used as the separation functional. The KL divergence is a natural distance function from a "true" probability distribution to a "target" probability distribution. Using the KL divergence, the separation functional of KCMAC-BYY, $F_s(M_{yang}, M_{ying})$ is given by

$$\sum_{(\mathbf{u}, z)} \int_{\mathbf{w}} p(\mathbf{w}|\ln \mu)p(\mathbf{u}|\mathbf{w})p(z|\mathbf{w}, \mathbf{u}) \ln \frac{p(\mathbf{w}|\ln \mu)p(\mathbf{u}|\mathbf{w})p(z|\mathbf{w}, \mathbf{u})}{p(\mathbf{u})p(z|\mathbf{u})p(\mathbf{w}|\mathbf{u}, z)} .$$

Considering the designed architecture, we have

$$\min_{M_{yang}, M_{ying}} F_s(M_{yang}, M_{ying}) \propto \max_{\mu, \xi} \ln \prod_{(\mathbf{u}, z)} p(\mathbf{u}, z|\ln \mu, \ln \xi) \tag{9}$$

From Eq. (9), minimizing the separation functional of the KCMAC-BYY is equivalent to maximizing $\prod_{(\mathbf{u}, z)} p(\mathbf{u}, z|\ln \mu, \ln \xi)$. It is noteworthy that, Suykens' work in LS-SVM [10] obtained a similar result. This situation not only validates the correctness of the novel KCMAC-BYY, but also provides a way to maximize the separation functional which is in common with that used by Suykens [10].

4.3 Hyperparameter Learning in the KCMAC-BYY

In practice, we can rewrite the optimization problem in μ and ξ into a scalar optimization problem in $\gamma = \xi/\mu$:

$$\max J_2(\gamma) = -\sum_{i=1}^{n-1} \ln \left[\lambda_{(G,i)} + \frac{1}{\gamma} \right] - (n-1) \ln \left(\frac{1}{2} \mathbf{w}_t^T \mathbf{w}_t + \gamma \frac{1}{2} \sum_{i=1}^n e_i^2 \right) \quad (10)$$

where $\lambda_{(G,i)}$ is the eigenvalues of Gram matrix, \mathbf{w}_t is the solution of the constrained optimization (1) with the optimal value γ_t of the last iteration. We can obtain the optimal hyperparameter γ by solving the above optimization problem using the quasi-newton method:

$$\gamma_{t+1} = \gamma_t + \Delta\gamma_t, \quad \Delta\gamma_t = -\alpha_t \mathbf{B}_t^{-1} \nabla f(\gamma_t) \quad (11)$$

where α_t is the step size, which is done to ensure that the Wolfe conditions are satisfied at each step of the iteration; $\nabla f(\gamma_t)$ is the gradient, \mathbf{B}_t is the Hessian, and

$$\nabla f(\gamma_{t+1}) = \left. \frac{\partial J_2}{\partial \gamma} \right|_{\gamma = \gamma_{t+1}} \quad (12)$$

$$\mathbf{B}_{t+1} = \mathbf{B}_t + \frac{y_t y_t^T}{y_t^T \Delta\gamma_t} - \frac{\mathbf{B}_t \Delta\gamma_t (\mathbf{B}_t \Delta\gamma_t)^T}{\Delta\gamma_t^T \mathbf{B}_t \Delta\gamma_t}, \quad y_t = \nabla f(\gamma_{t+1}) - \nabla f(\gamma_t) \quad (13)$$

5 Experimental Results

In this section, we compare the performance of the KCMAC-BYY with original kernel CMAC [3] and other versions of CMAC in regression. In our experiments, 1-D and 2-D *sinc* functions are approximated. The hardware configuration for our experiments is: CPU = Intel Pentium IV 2.66GHz, operating system = Microsoft Windows XP, memory available = 2 Gbytes.

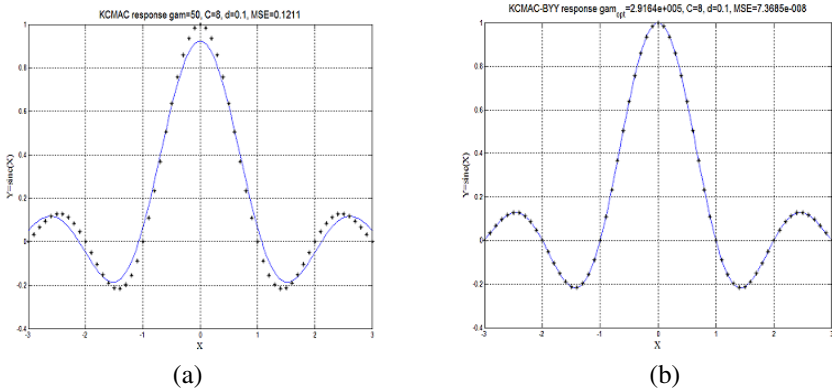


Fig. 3. Responses of the original KCMAC (a) and the KCMAC-BYY (b). Here the inputs are quantized into eight bits, while the number of levels, C , is set 8. The training data are taken from the interval $[-3, 3]$, and the distance between the neighboring samples is $d = 0.1$. Optimizing systematically the hyperparameter $\gamma = 2.9164 \times 10^5$, the KCMAC-BYY reduces the Mean Squared Error (MSE) to 7.3685×10^{-8} from 0.1211 in the KCMAC.

Similar results can be obtained in 2-D cases. In Table 1, the comparison of the root mean square (RMS) error of the KCMAC-BYY and other CMAC models for 2-D function approximation is given, where the inputs are sampled from the interval $[-5\pi, 5\pi]$ with different values of d . Here $C = 8$, and the input is quantized into 48×48 discrete values. Above experimental results demonstrate the KCMAC-BYY provide better performance than the other versions of CMAC in regression.

Table 1. Comparison of RMSE of *sinc* approximation using KCMAC-BYY and other CMACs

C=8	Training data	KCMAC-BYY	KCMAC	Albus CMAC	HCMAC	MS-CAMC	FCMAC-BYY
d=2	576	0.0021	0.0029	0.0092	0.0058	0.0092	0.0032
d=3	256	0.0124	0.0209	0.0233	0.0146	0.0233	0.0208
d=4	144	0.0119	0.0124	0.0166	0.0254	0.0166	0.0122
d=5	81	0.0495	0.0458	0.0480	0.0406	0.0480	0.0386
d=6	64	0.0329	0.0658	0.0737	0.0398	0.0737	0.0681

6 Conclusions and Future Work

The proposed KCMAC-BYY, using the Bayesian Ying-Yang learning theory, which is motivated by the well-known Chinese ancient Yin-Yang philosophy, achieves the systematic tuning of the hyperparameter, and further improves performance in modeling capability and stability. The experimental results show that the proposed KCMAC-BYY outperforms the existing representative techniques in the research literature. In future work, we will study an online approach to improve the proposed KCMAC-BYY, so that it will be able to handle online data as well as self-adapt during learning.

References

1. Albus, J.S.: Data storage in the cerebellar model articulation controller (CMAC). IEEE Transaction of the ASME, Dynamic Systems Measurement and Control 97(3), 228--233 (1975)
2. Ker, J.S., Kuo, Y.H., Wen, R.C., Liu, B.D.: Hardware implementation of CMAC neural network with reduced storage requirement. IEEE Transactions on Neural Networks 8(6), 1545--1556 (1997)
3. Horvath, G., Szabo, T.: Kernel CMAC with Improved Capability. IEEE Transactions on Systems, Man and Cybernetics Part B-Cybernetics 37(1), 124--138 (2007)
4. Xu, L.: BYY harmony learning, structural RPCL, and topological self-organizing on mixture models. Neural Networks 15, 1125--1151 (2002)
5. Xu, L.: Advances on BYY harmony learning: information theoretic perspective, generalized projection geometry, and independent factor auto determination. IEEE Transactions on Neural Networks 15, 885--902 (2004)

6. Xu, L.: RBF Nets, Mixture Experts, and Bayesian Ying-Yang Learning. *Neurocomputing* 19(1-3), 223--257 (1998)
7. Lane, S.H., Handelman, D.A., Gelfand, J.J.: Theory and development of higher-order CMAC neural networks. *IEEE Control System Magazine* 12(2), 23--30 (1992)
8. Chiang, C.T., Lin, C.S.: CMAC with general basis function. *Neural Networks* 9(7), 1199--1211 (1998)
9. Lee, H.M., Chen, C.M., Lu, Y.F.: A self-organizing HCMAC neural network classifier. *IEEE Transactions on Neural Networks* 14(1), 15--27 (2003)
10. Suykens, J.A.K., Gestel, T.V., Brabanter, J.D., Moor, B.D., Vandewalle, J.: *Least Squares Support Vector Machines*. World Scientific, Singapore (2002)
11. Nguyen, M.N., Shi, D., Quek, C.: FCMAC-BYY: Fuzzy CMAC Using Bayesian Ying-Yang Learning. *IEEE Transactions on Systems, Man and Cybernetics-Part B* 36(5), 1180--1190 (2006)
12. Devroye, L.: *A Course in Density Estimation*. Birkhauser Publisher, Boston (1987)
13. Nguyen, M.N., Shi, D., Fu, J.: An online Bayesian Ying-Yang learning applied to fuzzy CMAC. *Neurocomputing* 72, 562--572 (2008)
14. Kullback, S., Leibler, R.A.: On information and sufficiency. *Annals of Mathematical Statistics* 2(1), 79--86 (1951)

Closest Source Selection Using IVA and Characteristic of Mixing Channel

Choong Hwan Choi, Jae-Kwon Yoo, and Soo-Young Lee

Bio and Brain Engineering, KAIST,
335 Gwahangno, Yuseong-gu,
Daejeon 305-701,
Republic of Korea

imchwan@gmail.com, jaekweni@kaist.ac.kr, sylee@kaist.ac.kr

Abstract. This paper introduces a method for selecting a target source of interest. The target source is assumed to be the closest to sensors among all the other sources regardless of the target source not being the dominant power at the sensors. In this paper, we propose a simple method to select the closest source from signals separated by Independent Vector Analysis (IVA). The proposed method is processed in two-stages. Firstly, IVA is used to separate the mixed signals. Secondly, the mixing channel characteristics are used to choose the closest source. Simulated experimental results are presented to show how well the proposed method works.

Keywords: Blind Source Extraction, Blind Source Separation(BSS), Closest Source, Convolutional Mixture, Independent Vector Analysis.

1 Introduction

The process of separating mixed signals into original signals is known as blind source separation(BSS) and ICA is a particular case of BSS when sources are assumed independent. From the very beginning the main focus of this research has been to do just that, separate the signals. Convolutional ICA algorithms are introduced in order to solve the problems at hand with the main issue being the permutation problem. Many techniques have been proposed to solve it, which include smoothing the frequency domain filter [1,2], and the direction of arrival estimation [3]. Although these methods provided a good solution, additional algorithmic steps and computations were needed. The IVA method overcomes these problems and works well with any ill-posed geometric arrangements among sources and sensors [4].

All the separated sources may be important in some applications, such as EEG signals, but not in others. In some special applications such as selecting an order from multiple speech input, only one source is significantly important compared to the others. Sawada et al. has introduced a method to extract dominant target sources which are assumed to be close to the sensors [6]. The algorithm used in this method assumes the target sources have dominant powers at the sensors. So if the power of a signal far from the sensor is

larger than the closer signal, the algorithm does not work properly. Our proposed algorithm selects the source closest to the sensors whether or not the target source has the dominant power. Fig. 1 shows the flow of the proposed

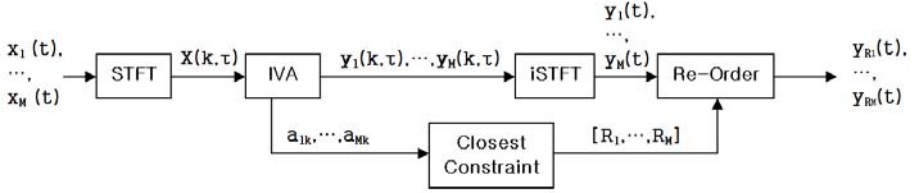


Fig. 1. Overall diagram of proposed method

method. Time-domain signals $x_i(t)$ are converted into time-frequency domain signals $x_i(k, \tau)$ by short-time Fourier transform (STFT). Then, we apply IVA to the signal $\mathbf{x}(k, \tau) = [x_1(k, \tau), \dots, x_M(k, \tau)]$ and obtain unmixing filter and separated sources $\mathbf{y}(k, \tau) = [y_1(k, \tau), \dots, y_M(k, \tau)]$. Basis vectors are used to select the closest source. Inverse STFT is applied to the separated sources $y_i(k, \tau)$ and we can obtain time-domain output signals $y_i(t)$. For the last step, we re-order the sequence of the output signals based on the closest constraint.

2 Independent Vector Analysis (IVA)

The first step is to separate the components of each sources. We apply IVA by assuming that the number of independent component is equal to M

$$\mathbf{y}(k, \tau) = \mathbf{W}_{::k} \mathbf{x}(k, \tau) . \tag{1}$$

where $\mathbf{W}_{::k} = [\mathbf{w}_{1k}, \dots, \mathbf{w}_{Mk}]^H$ is an $M \times M$ separation matrix of k th frequency bin.

In order to separate multivariate sources from multivariate observations, we define Kullback-Leibler divergence between two functions as the measure of independence. One is a total joint probability, $P(y_1, \dots, y_M)$ and the other is the product of marginal probabilities of individual source vectors, $\prod_{i=1}^M P(y_i)$. The object function that maximizes the independence of the output signal y_i can be written as

$$\begin{aligned} \mathbf{I}(\mathbf{y}) &= KL[P(Y) || \prod_{i=1}^M P(y_i)] \\ &= \int P(x_1, \dots, x_M) \log P(x_1, \dots, x_M) dx_1 \cdots dx_M \\ &\quad - \sum_{k=1}^K \log |\det W_{::k}| - \sum_{i=1}^M E[\log P(y_{i1}, \dots, y_{iK})] . \end{aligned} \tag{2}$$

$\int P(x_1, \dots, x_M) \log P(x_1, \dots, x_M) dx_1 \cdots dx_M$: is the entropy of the given observations, which is a constant. K is the number of frequency bins. The interesting parts of these cost functions are that each source is a multivariate and it is minimized when the dependency between the source vectors are removed but the dependency between the components of each vector does not need to be removed. Therefore, the object function preserves the inherent frequency dependency within each source, but it removes the dependency between the sources. By differentiating $\mathbf{I}(\mathbf{y})$ with respect to the coefficients of the separating matrices, w_{ijk} , we can obtain the gradients for the coefficients as following.

$$\Delta W_{ijk} \propto -\frac{\partial \mathbf{I}(\mathbf{y})}{\partial w_{ijk}} = \hat{h}_{jik}^* - E[\Phi_k(y_{i1}, \dots, y_{iK}) y_{jk}^*]. \tag{3}$$

where \hat{h}_{jik} is j th row and i th column element of the matrix $(W_{::k}^{-1})$ in k th frequency bin. \star denotes complex conjugate. And the nonlinear function, $\Phi_k(\cdot)$, is given as

$$\Phi_k(y_{i1}, \dots, y_{iK}) = -\frac{\partial \log P(y_{i1}, \dots, y_{iK})}{\partial y_{ik}}. \tag{4}$$

Note that the score function Φ_k is a multivariate function.

In our approach, we defined the complex-valued source distribution as a dependent multivariate super-Gaussian distribution that is spherically symmetric in all frequency bins. Most natural signals have inherent dependencies between frequency bins. Nonetheless, each frequency bin is uncorrelated with the others, because the Fourier bases are orthogonal bases. Thus, we can set the covariance term as a diagonal matrix. Since Fourier outputs have zero means, we can write the distribution as following.

$$P(y_i) = \alpha \cdot \exp\left(-\sqrt{\sum_k |y_{ik}|^2} \right). \tag{5}$$

where σ_{ik} is the variance of the i th source at the k th frequency bin, which determines the scale of each element of a source vector. In the algorithm, we set σ_{ik} to 1, because we adjust the scale after learning the results of the separating filters. Consequently, the multivariate score function we used is given as

$$\Phi_k(y_{i1}, \dots, y_{iK}) = \frac{\partial \sqrt{\sum_{k=1}^K |y_{ik}|^2}}{\partial y_{ik}} = \frac{y_{ik}}{\sqrt{\sum_{k=1}^K |y_{ik}|^2}}. \tag{6}$$

Although we used a fixed form of a multivariate score function as seen in the equation above, we do not claim that only this form is appropriate for separating source signals. Since the form of a multivariate score function is related to the dependency of sources, the proper form of a multivariate score function may vary with different types of dependency.

IVA avoids the permutation problem by exploiting the higher order frequency dependencies, but the scaling problem still need to be solved. If the sources are

stationary and the variances of the sources are known in all frequency bins, one can solve it by adjusting the variances to the known values. However, natural signal sources are dynamic, non-stationary in general, and the variances are unknown. Instead of adjusting the source variances, we can solve the scaling problem by adjusting the learned separating filter matrix. A well-known method is obtained by the minimal distortion principle [5]. Once the learning algorithm is finished, the learned separating filter matrix is an arbitrary scaled version of the exact one, which is given as

$$W_{::k} = D_{::k} H_{::k}^{-1} . \quad (7)$$

where $D_{::k}$ is an arbitrary diagonal matrix. Therefore, by replacing the separating filter matrix as,

$$W_{::k} = \text{diag}(W_{::k}^{-1}) W_{::k} . \quad (8)$$

where $\text{diag}(X)$ denotes the diagonal matrix which has same diagonal entries of the matrix X . After solving the scaling problem, we perform an inverse Fourier transform and overlap add in order to reconstruct the time domain signal. After IVA is performed, we can obtain not only separated source signals but also a separation matrix. We can obtain corresponding mixing channels for each sources from the separation matrix. By using the information of the mixing channels, we can select the closest source from the separated signals.

3 Selecting the Closest Source

By using the results of IVA, we can select the closest source to the sensors by a simple method. In this section, we discuss the characteristics of the closest mixing channel and the method used to select the closest source.

3.1 Characteristic of Mixing Channels

Original speech signals pass through the acoustic channels and are recorded by the microphones. Observed signal $x_j(t)$ from j th microphone can be represented as

$$x_j(t) = \sum_{i=1}^N x_{ji}(t), \quad j = 1, \dots, M . \quad (9)$$

where

$$x_{ji}(t) = \sum_{l=0}^{L-1} h_{ji}(l) s_i(t-l) . \quad (10)$$

$x_{ji}(t)$ is the component of s_i measured at sensor j . And $h_{ji}(l)$ is the impulse response from source i to sensor j and L is the filter length of the impulse response. The impulse response $h_{ji}(l)$ changes according to the relative position between source i and sensor j . An impulse response of the source close to a sensor looks more like a delta function. The delta function in the time domain is equal to the uniform function in the frequency domain. So if we can approximate a good mixing channel, then we can select which source is the closest to the sensors.

3.2 Selecting the Closest Source

From the IVA result, we can get the unmixing matrix $\mathbf{W}_{::k}$ for all frequency bins and separated signals $y_i(k, \tau)$. In this case we only consider the case of equal number of sources and sensors. So to obtain the mixing matrix, we calculate the inverse of $\mathbf{W}_{::k}$

$$[\mathbf{a}_{1k}, \dots, \mathbf{a}_{Mk}] = \mathbf{W}_{::k}^{-1}, \quad \mathbf{a}_{ik} = [a_{1ik}, \dots, a_{Mik}]^T. \quad (11)$$

By multiplying both sides of Eq. 11 by $\mathbf{W}_{::k}^{-1}$, the sample vector $\mathbf{x}(k, \tau)$ is represented by a linear combination of basis vectors $\mathbf{a}_{1k}, \dots, \mathbf{a}_{Mk}$

$$\mathbf{x}(k, \tau) = \sum_{i=1}^M \mathbf{a}_{ik} y_i(k, \tau). \quad (12)$$

If y_i is well trained to be the original source s_i , \mathbf{a}_{ik} can be one to one matching to $\mathbf{h}_{ik} = [h_{1ik}, \dots, h_{Mik}]^T$ where h_{jik} is the frequency response from source i to sensor j at k th frequency bin. Since we know that \mathbf{a}_{ik} is the mixing basis from the i th source to each sensors, we calculate the flatness F_i of each basis as following.

$$F_i = \sum_{j=1}^M \frac{\text{var}_k(a_{jik})}{\|\text{mean}_k(a_{jik})\|}. \quad (13)$$

The separated source with smallest flatness value is the closest source. Then we renumber the indexes of the separated components $y_1(k, \tau), \dots, y_M(k, \tau)$ accordingly that the closest source comes to the first.

4 Experiments

4.1 Experimental Setting and Performance Measures

We performed experiments to select the target source which was the closest to the microphone arrays. Simulated room environments are shown in Fig. 2. We

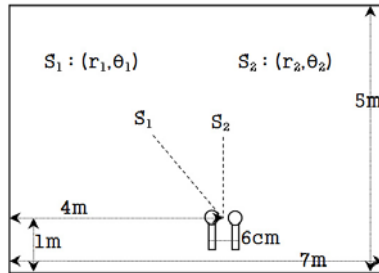


Fig. 2. Simulated room environments

set the room size to 7m x 5m x 2.75m and set all heights of the microphone and source locations to 1.5m. A reverberation time was 50ms and the corresponding reflection coefficients were set to 0.57 for every wall, floor, and ceiling.

The mixed speech signals we used were recorded speech signals generated in a simulated room environment. Recorded speech signals were clean speech signals 8 seconds long sampled at 8kHz, and they were convolved with corresponding room impulse response. 1024-point FFT and a 1024-tab long Hanning window with the shift size of 256 samples were used.

Our proposed methods were applied to 2 x 2 speech separation problems. Two array microphones are located in the middle of the room. The position of the source signal S_i is represented by the distance and the angle(r_i, θ_i) from the center of the microphone array. The separation performance was measured by the signal to interference ratio in dB defined as

$$SIR_s = 10\log\left(\frac{\sum_{t,k} |s_2^k[t]|^2}{\sum_{t,k} |s_1^k[t]|^2}\right). \quad (14)$$

$$SIR_x = \frac{1}{M} \sum_i^M 10\log\left(\frac{\sum_{t,k} |h_{i2}^k s_2^k[t]|^2}{\sum_{t,k} |h_{i1}^k s_1^k[t]|^2}\right). \quad (15)$$

$$SIR_y = 10\log\left(\frac{\sum_{t,k} |r_{q(2)2}^k s_2^k[t]|^2}{\sum_{t,k} |r_{q(1)1}^k s_1^k[t]|^2}\right). \quad (16)$$

where $q(i)$ indicates separated source index that the i th source appears, and $r_{q(i)j}^k$ is an overall impulse response, which is defined by $\sum_m w_{q(i)m}^k h_{mj}^k$ at k th frequency bin.

4.2 Two Sources at a Same Direction

First, the proposed algorithm was applied to the problem with two sources located at the same direction. The results are shown in Table. [1](#). The experiments were done with various directions. Source signal S_2 was defined as the target signal. The goal of proposed algorithm is to select S_2 as the closest source. S_2 was well selected for when the other signal was at the same direction in most of the cases. The angle between 10° and 20° failed to select the correct closest source. This is because the performance of the proposed algorithm depends on the result of the separating algorithm. This experimental setup is difficult to

Table 1. Two speakers are at same direction

$S_1 : (r_1, \theta_1)$	$S_2 : (r_2, \theta_2)$	SIR_s (dB)	SIR_x (dB)	SIR_y (dB)	IVA	Proposed Method
(2m, 0°)	(1m, 0°)	0	5.77	18.01	S_1	S_2
(2m, 10°)	(1m, 10°)	0	5.87	-25.90	S_1	S_1
(2m, 20°)	(1m, 20°)	0	5.84	-26.41	S_1	S_1
(2m, 30°)	(1m, 30°)	0	6.03	18.96	S_1	S_2
(2m, 45°)	(1m, 45°)	0	5.38	18.90	S_1	S_2

Table 2. Target source S_2 is positioned at (r_2, θ_2) . Other source is at same direction with different distance.

$S_1 : (r_1, \theta_1)$	$S_2 : (r_2, \theta_2)$	SIR_s (dB)	SIR_x (dB)	SIR_y (dB)	IVA	Proposed Method
(1.1m, 0°)	(1m, 0°)	0	0.92	10.60	S_1	S_2
(1.1m, 0°)	(1m, 0°)	-10	-9.08	3.61	S_2	S_2
(1.7m, 0°)	(1m, 0°)	-15	-10.28	2.97	S_2	S_2

Table 3. Target source S_2 is positioned at (r_2, θ_2) . Other source is at different distance with various angles. Experiments are done with different SIRs.

$S_1 : (r_1, \theta_1)$	$S_2 : (r_2, \theta_2)$	SIR_s (dB)	SIR_x (dB)	SIR_y (dB)	IVA	Proposed Method
(2m, -60°)	(1m, 0°)	0	5.66	30.29	S_1	S_2
(2m, 10°)	(1m, 0°)	0	5.81	20.51	S_1	S_2
(2m, -60°)	(1m, 0°)	-10	-4.34	22.58	S_2	S_2
(2m, 0°)	(1m, 0°)	-10	-4.25	9.72	S_2	S_2
(2m, 10°)	(1m, 0°)	-10	-4.19	13.59	S_2	S_2
(2m, -60°)	(1m, 0°)	-20	-14.34	11.05	S_2	S_2
(2m, -25°)	(1m, 0°)	-20	-14.16	9.91	S_2	S_2
(2m, 25°)	(1m, 0°)	-20	-14.17	9.76	S_2	S_2
(2m, 40°)	(1m, 0°)	-20	-14.38	7.01	S_2	S_2

solve, because the sources are located closely together on the same side and have the same direction of arrival(DOA). Though the IVA succeed to separate each sources, the unmixing filter estimation is still hard to solve and it may cause some problems. If the separation method cannot estimate exact unmixing filter, the proposed method also fails to select the correct target source. Except for the angles between 10° and 20° , the proposed method works well.

4.3 Refine Experiments

Next, we show the experiment results for the refinement. Although we have performed experiments under various conditions, here we simply show the results for some cases. Second experiment results are shown in Table. 2. The position of target source S_2 is fixed to $(1m, 0^\circ)$. Though the target source power is not dominant to the sensors, the proposed algorithm can select the correct closest source. The results from the third experiment are shown in Table. 3. When SIR_s is 0 or -10, the system selected the correct answer wherever S_1 was positioned. For the case of SIR_s equals to -20, the system fails to select the closest source if the direction of two sources are similar.

5 Conclusions

We have proposed a simple method for selecting the target source from the mixed signals. The target source is defined as the closest source from the sensors.

The mixing channel characteristics were used to select the target source. The impulse response from the source to sensor is more like an impulse function as the source is closer to sensor. It means the frequency response is more uniform in the frequency domain. Although the power of target source to the sensor is relatively small, the proposed method choose the closest source very well.

References

1. Smaragdis, P.: Blind Separation of Convolved Mixtures in the Frequency Domain. *Neurocomputing* 22, 21–34 (1998)
2. Parra, L., Spence, C.: Convolutional Blind Separation of Non-Stationary Sources. *IEEE Trans. Speech and Audio Processing* 8(3), 320–327 (2000)
3. Sawada, H., Mukai, R., Araki, S., Makino, S.: A Robust and Precise Method for Solving the Permutation Problem of Frequency-Domain Blind Source Separation. In: *Proceedings of International Conference on ICA and BSS*, pp. 505–510 (2003)
4. Lee, I., Kim, T., Lee, T.-W.: Fast Fixed-Point Independent Vector Analysis Algorithms for Convolutional Blind Source Separation. *Signal Processing* 87, 1859–1871 (2007)
5. Matsuoka, K.: Minimal Distortion Principle for Blind Source Separation. In: *Proceedings of the 41st SICE Annual Conference*, vol. 4, pp. 2138–2143 (2002)
6. Sawada, H., Araki, S., Mukai, R., Makino, S.: Blind Extraction of Dominant Target Sources Using ICA and Time-Frequency Masking. *IEEE Transactions on Audio, Speech, and Language Processing* 14(6), 2165–2173 (2006)

Decomposition Mixed Pixels of Remote Sensing Image Based on 2-DWT and Kernel ICA

Huaiying Xia and Ping Guo

Laboratory of Image Processing and Pattern Recognition,
Beijing Normal University Beijing, 100875, China
xia_huaiying@163.com, pguo@ieee.org

Abstract. In this paper, we propose a novel method for decomposing mixed-pixels of remote sensing images, which integrates two-Dimensional Wavelet Transform (2-DWT) and Kernel Independent Component Analysis (KICA) technique. In order to improve the signal and noise ratio of the original mixed-pixel images, we apply wavelet analysis method to reduce the noise of the images. High-frequency sub-image in wavelet domain is approximately represented by a kind of super-Gaussian Laplace distribution, and KICA is adopted for this distribution with greater kurtosis for obtaining higher accuracy and faster convergence rate. The experiments show that decomposition result with the proposed method is much improved not only at accuracy but also remarkably robust to noise compared those obtained with 2-DWT-ICA or KICA.

Keywords: Wavelet Transform, Kernel Independent Component Analysis, Remote Sensing Image, Mixed Pixel.

1 Introduction

Due to restrictions on spatial resolution of sensors, the complexity of the diversity of features, as well as the impact factors such as the atmosphere, topography, surface features, the second reflection, mixed pixels in remote sensing images exist widely. Mixed pixels refer to the pixels that cover more than one constituent material within the instantaneous field of view (IFOV) of the sensor [1]. The decomposition of mixed pixels is a growing research area with a wide range of applications such as sub-pixel object quantification, mineral identification, area estimation [2], etc. Another emerging application developed recently in biological microscopy is to analyze multi-spectral fluorescence microscopy for discriminating different co-localized fluorescent molecule [3].

Many techniques for decomposition of mixed pixels for remote sensing images have been developed in recent years, these techniques can mainly be categorized into linear or nonlinear mixture model [4]. The linear mixture model has gained significant popularity due to its effectiveness and simplicity. In recent years, most decomposition methods have been developed based on linear mixed models, which can be classified into three groups based on their specific focuses.

The first group of algorithms focuses on end-member extraction. Various techniques have been investigated, including spectral angle mapper, projection pursuit, convex hull geometry [5]. The second group performs abundance estimation if end-member signatures are given, including maximum-likelihood estimation, linear mixture analysis, fuzzy c-means [4] and artificial neural networks [2]. The third group of methods attempts to take account of both procedures, using either least squares or blind source separation methods, such as Independent Component Analysis (ICA) [6], and Non-negative Matrix Factorization(NMF) [7].

The traditional ICA has some limitations in mixed-pixel decomposition since its linear properties, while kernel ICA (KICA) has a nonlinear kernel mapping, and could be adopted to improve the decomposition task under the circumstance of Gaussian distribution [6,7,8,9]. However, the original remote sensing images usually contain noise and signal to noise ratio(SNR) is low, applying KICA directly can not obtain desired effects. Then we propose a novel mixed pixel decomposition method which adopts Two-Dimensional Wavelet transform (2-DWT) analysis and KICA technique, termed 2-DWT-KICA, to improve the un-mixing pixel task in this paper.

2 Background

In this section, we briefly review the relative mathematic theory and background knowledge.

2.1 Linear Model of Mixed Pixel

Assuming that there are m remote receivers, observation vector of each pixel is $\mathbf{x} = (x_1, x_2, \dots, x_m)^T$. The basic assumption of linear mixing is that within the IFOV [1] of a single pixel is given by,

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n}, \quad x_i = \sum_{j=1}^c a_{ij}s_j + n_i, \quad (1)$$

where \mathbf{A} is an $m \times c$ reflection coefficient matrix, whose column, $a_j, j = 1, \dots, c$, corresponds to the spectral signature of the j th endmember, and c is the number of endmembers. The abundance vector is denoted by \mathbf{s} , which satisfies two physical constraints, $s_j \geq 0$ and $\sum_{j=1}^c s_j = 1$. Noises are taken into account by an $m \times 1$ column vector \mathbf{n} .

The target of decomposition is to find a matrix \mathbf{W} , when applied to the observation \mathbf{x} , the true abundance can be gained $\tilde{\mathbf{s}} = \mathbf{W}\mathbf{x}$ according to some algorithm [1].

2.2 Kernel Independent Component Analysis

The main idea of ICA is in search of a non-singular transformation for multivariate data and make the transformed component is independent to each other as much as possible. More details about ICA algorithm can be found in [6]. As

we known, ICA is a linear method, most of data is nonlinear distribution which is too complex to be presented well by a linear model in practice. Harmeling et al [10] presented a kernel-based algorithm in the blind separation of nonlinearly mixed speech signals. One of the eye-catching characteristics of Kernel function method is that it can use Mercer kernel functions in place of the linear inner product algorithm to achieve nonlinear transformation without considering the specific form of non-linear transformation [8]. The basic idea of their algorithm is to map the input data into an implicit feature space \mathbf{F} with kernel trick firstly:

$$x \in \mathbf{R}^N \rightarrow \Phi(x) \in \mathbf{F} . \tag{2}$$

then ICA is performed in \mathbf{F} to produce a set of nonlinear features of input data. Selecting an appropriate kernel function for a particular application area is very important. Many functions can be chosen for the kernel such as Polynomial kernel, Gaussian kernel , sigmoid kernel, cosine kernel and so on. In this paper, we will adopt Gaussian Kernel [9]. A brief introduction of the algorithm is given in the following:

Input Data: Observation vector \mathbf{x} and Kernel function. Gaussian kernel function which is defined as expression (3):

$$k'(x, y) = \exp(-(x - y)^2/2\sigma^2) . \tag{3}$$

Step 1: Whitened the input data. The whitening matrix is:

$$\mathbf{X}_{\Phi}^W = (\mathbf{W}'_{\Phi})^T \Phi(\mathbf{X}) = (\Lambda_{\Phi})^{-1} \alpha^T \mathbf{K} , \tag{4}$$

where \mathbf{K} is defined by $k_{ij} = \Phi(\mathbf{x}_i)\Phi(\mathbf{x}_j)$. The kernel function k can be computed instead of Φ . Λ_{Φ} is the eigenvalue matrix of covariance matrix of \mathbf{X} . α is the eigenvector matrix of \mathbf{K} .

Step 2: Compute \mathbf{W} according to expression (4) by the following iterative algorithm:

$$\mathbf{Y}'_{\Phi} = (\mathbf{W}_{\Phi} \mathbf{X}_{\Phi}) . \tag{5}$$

$$\Delta \mathbf{W}_{\Phi} = [\mathbf{I} + (\mathbf{I} - \frac{2}{1 + e^{-\mathbf{Y}'_{\Phi}}}) (\mathbf{Y}'_{\Phi})^T] \mathbf{W}_{\Phi} . \tag{6}$$

$$\mathbf{W}'_{\Phi} = \mathbf{W}_{Phi} + \rho \Delta \mathbf{W}_{\Phi} \rightarrow \mathbf{W}_{\Phi} . \tag{7}$$

Repeated the step 2, until \mathbf{W}_{Phi} converged, where ρ is a learning constant.

Output: \mathbf{W}

2.3 Two-Dimensional Wavelet Transform

Wavelet transform can be applied to analyze the local characteristics of signal both in the time domain and frequency domain. Two-dimensional wavelet transform was applied to decompose two-dimensional signal of images into high-frequency and low frequency information in the following experiment. Universal threshold algorithm [11] was chosen to select the wavelet coefficient. The

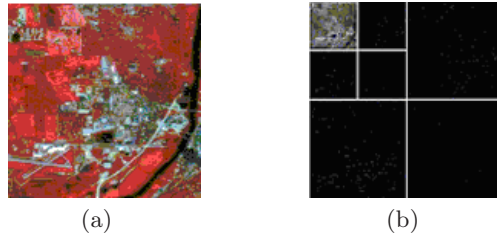


Fig. 1. Wavelet Decomposition, (a) Original pseudo-color image of the AVIRIS data, and (b) Sub-images of low frequency, horizontal, vertical and diagonal direction in the wavelet domain after two wavelet decomposition

coefficient of low-frequency mainly reflects the information of signal, and the coefficient of high-frequency mainly reflects the noise and the details of signal.

The feature distributions of most of images are Gaussian distribution. High-frequency sub-image in wavelet domain is approximated for the Laplace distribution which is a kind of super-Gaussian distribution with greater kurtosis [11]. The original image and its four sub-images in wavelet domain (low frequency, horizontal, vertical and diagonal directions, respectively) are shown in Fig. 1, and Table 1 shows the kurtosis of those sub-image in wavelet domain. Normalized kurtosis is defined as follows [11]:

$$kurt(x) = \frac{E\{(x - X')^4\}}{[E\{(x - X')^2\}]^2} = E\{x\}. \tag{8}$$

As shown in Table 1, the kurtosis of high-frequency sub-image is increased nearly 10 times. Other sub-images have the similar property.

3 Mixed Pixel Decomposition Based on 2-DWT-KICA

Following the analysis from above, the steps of performing the proposed method can be summarized as follows:

- a) A wavelet and the layer of decomposition are selected first. Then two-dimensional wavelet transform is applied to mixed images.
- b) High-frequency sub-image is chosen, and stack it into a matrix \mathbf{X}' according its row.
- c) Kernel ICA is applied to matrix \mathbf{X}' , and then the separation matrix \mathbf{W} can be calculated by iterative methods described in subsection 2.2.

Table 1. The kurtosis of low frequency, high-frequency, vertical and diagonal direction sub-image in wavelet domain

	Original	Low	Horizontal	Vertical	Diagonal
kurt	2.14	2.19	20.7	19.8	19.6

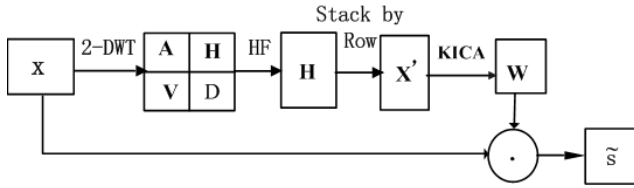


Fig. 2. Flowchart of the proposed method

d) Estimated signal source can be calculated by the expression: $\tilde{\mathbf{s}} = \mathbf{W}\mathbf{x}$, \mathbf{x} is the mixed image matrix in time-domain.

e) Finally, restore every row of the estimation to the form of two-dimensional images to obtain the independent source images. Fig. 2 shows the flowchart of the proposed method.

4 Experiments

In order to verify the effectiveness of the proposed method, simulated images as well as a real TM remote sensing images are used in the experiment. Both 2-DWT-ICA and KICA are compared with proposed method also. The programs are run on the platform of PC with CPU 2.4Ghz.

4.1 Experiments with Simulated Images

In this part, synthetic images are generated based on the following steps:

- Generate mixing coefficients matrix \mathbf{s} with 128×128 pixels randomly and satisfy two physical constraints $s_j \geq 0$ and $\sum_{j=1}^c s_j = 1$.
- Generate reflection coefficient matrix \mathbf{A} randomly with the size of 128×128 .
- Calculate the observation vector: $\mathbf{x} = \mathbf{A}\mathbf{s}$.
- Add zero mean Gaussian random noise to \mathbf{x} .

In the experiment, the proposed method, 2-DWT-ICA, and KICA are tested with simulated image, respectively. We have chosen different direction high-frequency sub-image of mixed images and different wavelet such as harr, bior4.4, coif3 and sym4. After comparison, we found that horizontal sub-image and harr wavelet has best separation accuracy. So the horizontal sub-image and harr wavelet are chosen in the following experiment. Fig. 3 is the simulated image and its' decomposition results using 2-DWT-KICA.

Table 2 shows the proportion of four mixed pixels chosen randomly from the simulated mixed image. From Table 2, we can see that the decomposition results of the proposed method is approximate to the ratio of the real surface features.

Further, the root-mean-square error (RMSE) and correlation coefficient (CC) are used to evaluate the performance of the algorithms [11]. The smaller RMSE

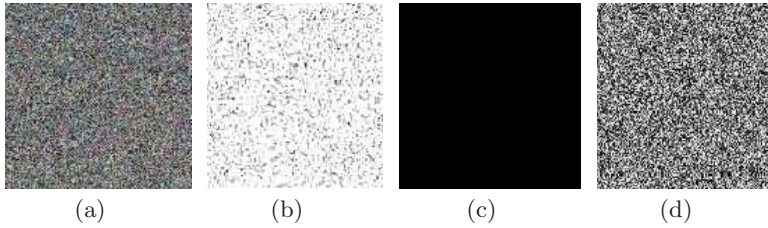


Fig. 3. Experiments with Simulated Images. (a) Simulated mixed image, (b), (c) and (d) are decomposition results using the proposed method.

Table 2. The accuracy comparisons of different methods

	Original Ratio	The Proposed Method			KICA			2-DWT-ICA		
1	0.20 0.40 0.50	0.244	0.467	0.581	0.250	0.501	0.594	0.322	0.385	0.687
2	0.12 0.30 0.58	0.122	0.358	0.601	0.157	0.450	0.625	0.223	0.564	0.684
3	0.25 0.35 0.40	0.210	0.406	0.451	0.202	0.415	0.487	0.307	0.438	0.560
4	0.16 0.34 0.50	0.142	0.405	0.562	0.164	0.421	0.616	0.283	0.265	0.754

and greater CC represent the higher accuracy of the decomposition results. More details can be found in [11]. Table 3 shows the results of these two indicators of decomposition with different methods.

Table 2 and Table 3 show the unmixing results under additional Gaussian noise. We can see that our method is more robust to noise and has higher decomposition accuracy than 2-DWT-ICA and KICA.

4.2 Experiments with TM Remote Sensing Image

For the lack of standard criterion for real remote sensing images, we cannot evaluate the algorithm performance by objective numerical values. Here the only way to evaluate the algorithm performance is by comparing the visual effects. The image with the size of 128×128 used in the experiment, which is downloaded from the website [12]. This image is often used for testing the performance of un-mixing algorithm. Original image is in Fig. 1(a), as the ground truth in [12] shows, the image consists of three major types of features: natural vegetation, artificial structures and hay. Images shown in Fig. 4(a), (b) and (c) are

Table 3. The accuracy indexes and different methods

	$RMSE_1$	$RMSE_2$	$RMSE_3$	CC_1	CC_2	CC_3
2-DWT-KICA	0.1167	0.1135	0.1573	0.9435	0.9357	0.9287
2-DWT-ICA	0.1516	0.2132	0.2098	0.8705	0.8826	0.8559
KICA	0.1321	0.2018	0.1825	0.9124	0.8932	0.9024

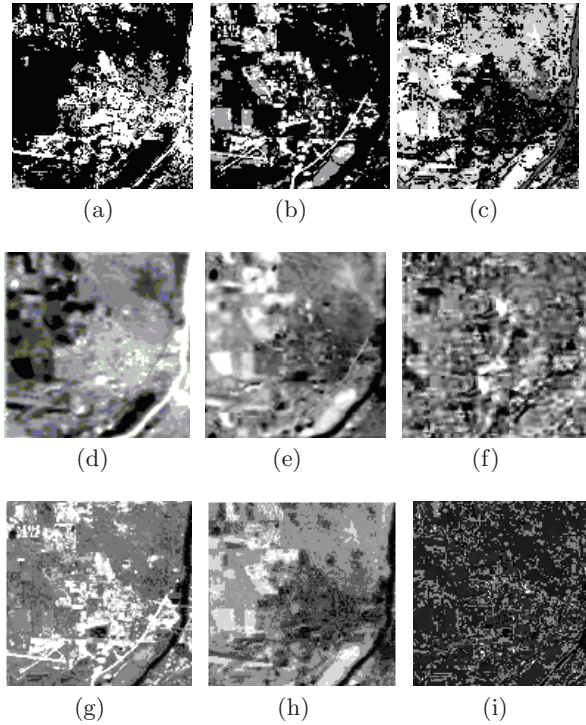


Fig. 4. Experiments with TM remote sensing image. (a), (b), and (c) are decomposition results of the proposed method. (a) Natural Vegetation, (b) Artificial Structures, and (c) Hay. (d), (e), and (f) are decomposition results of applying KICA. (d) Natural Vegetation, (e) Artificial Structures, and (f) Hay. (g), (h), and (i) are decomposition results of 2-DWT-ICA. (g) Natural Vegetation, (h) Artificial Structures, and (i) Hay.

the decomposition results of applying proposed algorithm. We can see that the proposed method can separate mixed pixels quite well. The profile of natural vegetation, artificial structures and Hay are clearly visible. Fig. 4(d), (e) and (f) show the results using KICA algorithm. It is obvious that the decomposition results are affected by noise seriously. Fig. 4(g), (h), and (i) show the decomposition results with 2-DWT-ICA method. The profile of natural vegetation is blurred, and the details of hay was vague.

Above experimental results show that the proposed method has a better visual effect than both KICA and 2-DWT-ICA, besides, it is robust to noise.

5 Conclusions and Future Work

In this paper, we proposed a new mixed pixel decomposition method which integrates Two-Dimensional Wavelet transform and Kernel Independent Component

Analysis techniques. The results showed that the proposed method has high decomposition accuracy as well as robustness to noise, and it is an effective solution for decomposing mixed pixels.

The proposed method still deserves further study. Selecting an appropriate kernel function for a particular application area can be difficult and remains largely an unresolved issue.

Acknowledgments. The research work described in this paper was fully supported by the grants from the National Natural Science Foundation of China (Project Nos. 60675011, 9082001). Prof. Ping Guo is the author to whom all correspondence should be addressed.

References

1. Miao, L., Qi, H., Szu, H.: A Maximum Entropy Approach to Unsupervised Mixed-pixel Decomposition. *IEEE Trans. on Image Proc.* 16(4), 1008–1021 (2007)
2. Wang, Z.H., Hu, G.D., Yao, S.Z.: Decomposition Mixed Pixel of Remote Sensing Image Based on Tray Neural Network Model. In: Kang, L., Liu, Y., Zeng, S. (eds.) *ISICA 2007*. LNCS, vol. 4683, pp. 305–309. Springer, Heidelberg (2007)
3. Liu, L.F., Wang, B., Zhang, L.M.: Decomposition of Mixed Pixels Based on Bayesian Self-organizing Map and Gaussian Mixture Model. In: *International Conference on Intelligent Computing Theory and Methodology*, vol. 30(9), pp. 820–826 (2008)
4. Keshava, N.: A Survey of Spectral Un-mixing Algorithms. *J. Lincoln Laboratory.* 14(1), 55–78 (2003)
5. Nascimento, J.M.P., Dias, J.M.B.: Vertex Component Analysis: A Fast Algorithm to Unmix Hyperspectral Data. *IEEE Trans. on Geo-sci. & Remote Sen.* 43(4), 898–910 (2005)
6. Nascimento, J.M.P., Dias, M.B.: Does Independent Component Analysis Play a Role in Un-mixing Hyper-spectral data. *IEEE Trans.on Geo-sci. & Remote Sen.* 43(1), 175–184 (2004)
7. Robila, S.A., Maciak, L.G.: Considerations on Parallelizing Non-negative Matrix Factorization for Hyper-spectral Data Un-mixing. *IEEE Geo-sci. & Remote Sen. Letters.* 6(1), 57–61 (2009)
8. Xu, A.B., Jin, X., Guo, P., Bie, R.F.: KICA Feature Extraction in Application to FNN based Image Registration. In: *Proceedings of 2006 International Joint Conference on Neural Networks*, vol. 10, pp. 3602–3608 (2006)
9. Bai, L., Xu, A.B., Guo, P., Jia, Y.D.: Kernel ICA Feature Extraction for Spectral Recognition of Celestial Objects. In: *Proceedings of 2006 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 5, pp. 3922–3926 (2006)
10. Harmeling, S., Ziehe, A., Kawanabe, M., Muller, K.R.: Kernel-based Nonlinear Blind Source Separation. *Neural Computation* 15(5), 1089–1124 (2003)
11. Chen, Y., He, Y., Zhu, X.H.: ICA Based Wavelet Transform and Its Application to Image Separation. *Modern Electronics Technique* 24, 131–134 (2007)
12. A Freeware Multispectral Image Data Analysis System,
<http://cobweb.ecn.purdue.edu/~biehl/MultiSpec/>

Echo Energy Estimation in Active Sonar Using Fast Independent Component Analysis

Dongmin Jeong¹, Kweon Son², Yonggon Lee², and Minho Lee¹

¹ School of Electrical Engineering and Computer Science
Kyungpook National University, 1370 Sankyuk-Dong, Puk-Gu, Taegu 702-701 Korea

² Agency for Defense Development, Jinhae P.O. Box 18, Kyungnam, 645-600 Korea
dmjeong@ee.knu.ac.kr, {sk142298,yongon}@add.re.kr,
mh01ee@knu.ac.kr

Abstract. In many underwater applications, it is desirable to separate independent signals according to their sources, allowing targets to be distinguished from self-noise, ambient noise and clutter. The long-term goal of this work is to better detect and model target echo under several location in real-ocean environments, and to develop signal processing techniques for echo energy estimation. This paper addresses echo energy estimation problem of active sonar in a set of sensors. This may be done by measuring a noiseless source signal echoed by a target whose acoustic properties are known. We propose an echo energy estimation method based on the following two stages; One is the blind source separation using an independent component analysis (ICA) to separate the remaining mixture into its independent components. We use the principal component analysis (PCA), as a preprocessor, to increase the input signal-to-noise ratio (SNR) of the succeeding ICA stage and to reduce the sensor dimensionality, and followed by the fast Fourier transform (FFT). As the second, after finding an original target echo signal, the energy estimation solution is newly proposed by considering an inverse procedure of the first stage, where the estimated sonar source is used as input for the pseudo-inverse procedure of the ICA filter combined with PCA. Then, we can estimate noise-free energy information of a target echo, which is compared with conventional beam forming method. The real-ocean recorded data demonstrate the performance of the proposed algorithm.

Keywords: Independent component analysis (ICA), active sonar system, echo energy estimation, signal separation.

1 Introduction

Systems employing the sound in underwater environment are known as sonar systems. Sonar, and an acronym for Sound Navigation and Ranging is a system used for the detection of objects [1, 2]. It is effective in underwater environment, and that is due to water's ability to propagate sound efficiently. Active sonar creates pulse of sound, often called a "ping", and then listens for reflections (echo) of the pulse. The pulse may be at constant frequency or a chirp of changing frequency. In this paper, we use this target echo in active sonar.

The problem of target echo estimation has received considerable attention in the last few years. Many papers [3-5] have been published on the location of a target, by estimating the direction of arrival of a signal emitted by the target. However, little attention has been focused on the high-accuracy identification of targets by measuring time-delay and the Doppler companding factor in an active system, for instance, radar and active sonar. Moreover, there are many complicated noises in the ocean and it is very difficult work to estimate accurately the underwater target movement. So, in underwater applications, it is desirable to separate source signals to be distinguished from self-noises, ambient noises, clusters and so on for estimating target movement. Beam forming methods [6] which existing is representative to estimate energy are signal processing techniques used in sensor arrays for directional signal transmission or reception. This spatial selectivity is achieved by using adaptive or fixed receive/transmit beam patterns. It has found numerous applications in radar, sonar, seismology, wireless communications, radio astronomy, speech and biomedicine [6]. Adaptive beam forming is used to detect and estimate the signal-of-interest at the output of a sensor array by means of data-adaptive spatial filtering and interference rejection. But this method is a limit of removing various noises in real-ocean environment.

Recently, blind source separation, or BSS, within the framework of independent component analysis (ICA) has attracted a great deal of attention in engineering field [7]. It has been widely noticed that there are many possible applications such as removing additive noises from signals and images, separating crosstalk in telecommunication, preprocessing for multi-probed radar-sonar signals. The ICA algorithm is the problem to separate independent sources given a mixed signal where the mixing process is unknown [8, 9]. We want to extract each source from the mixed signals using some techniques. Even if the mixing process is unknown, we can separate the sources if they are independent to each other [10]. The major approaches of blind source separation use higher order statistics but not the temporal structure of input signals. These algorithms also need iterative calculations for estimating the source signals because in most cases, they require non-linear optimization. This problem is encountered in the field of acoustics when M source signals of superposition are recorded by N microphones in a reverberant environment. We can separate and estimate the source signal from several noises using this scheme. The aim of this study is to estimate the energy of a reconstructed target echo. The source i.e. target echo will be assessed as omnidirectional, characteristic of a simple monopole point source, and there are many components which are misunderstood as the source signal. So, it is very important work to separate signals and estimate the accurate energy of the target echo.

This paper is organized as follows: in section 2, we describe some basic approaches to blind source separation of mixed signals and explain the experimental environment. In section 3, we propose a new energy estimation algorithm based on blind source separation of active sonar signals. In section 4, computation and experimental results of the proposed algorithm will be shown. Finally, we give brief summary and conclusion remarks in section 5.

2 Problem Description

There are many noises as the warship, the merchant ship and the clipper etc. various vessel flow in the ocean. In addition, the underwater vehicle makes not only self noises but also various environmental noises according to movement. So, it is very difficult work to estimate energy of target echo and detect the underwater object's location and movement. Also, target echo from a sonar system is occurred with various direction and stored reverberation signals other than target echo in the sensor array. Thus, it is necessary not only to separate source signal from noises but also estimate accurate arriving time of target echo. We assume that we know the ping generation and arriving times on a target.

First, we use the blind source separation (BSS) to determine source signals that form a mixture. There is the assumption that each component of source signals $s(t)$ is independent of each other and observations $x(t)$ correspond to the recorded signals. In the basic blind source separation problem, we assume that observations are linear mixtures of source signals:

$$x(t) = As(t), \quad (1)$$

where A is an unknown linear operator. A typical example of linear operators is a $m \times n$ real valued matrix. This formulation represents non-delayed (instantaneous) linear mixing [7]. Given the N linearly mixed input signals, we need to recover the M statistically independent sources as much as possible ($N \geq M$). The goal of blind source separation is to find a linear operator W such that the components of the reconstructed signals

$$y(t) = Wx(t) \quad (2)$$

are mutually independent, without knowing operator A and the probability distribution of source signal $s(t)$. Ideally we expect W to be the inverse of operator A , but since we lack of information about the amplitude of the source signals and their order, their remains indefiniteness of permutation and dilation factors [6]. Then, we can calculate the arriving time of target echo from distance and extract the target echo range for reconstructing the pure target echo. We can calculate the ping delivery time (T) from measuring a delivery time from the signal receivers and detection time of the target. From this, we can calculate the relative distance (R) from sensor to target as

$$R = T \times S, \quad (3)$$

where S is the average speed of sound. Finally we can extract the target echo range from raw data in the sensor array. After then we can separate the target echo from recorded mixtures of sensor array.

Figure 1 demonstrates the method of acquisition data in a real ocean. We use the LFHUSS (low-frequency and high-power underwater sonar system) as source and 1.5m length 10 channels @7kHz sonar sensor array. There are two ships for experiment. One has the 10 channels sensor array and the other has the echo repeater. They

maintain the relative distance which is fixed during experiment. The sensor array is located at 42m from a test ship and the echo repeater is also located at similar position. The echo repeater can store and amplify the target echo. So, we can estimate the energy increment and decrement of a target echo. There are two recorded data which are Arr data and ER data. The Arr data are the transmission ping signal and target echo signal, i.e. 1 and 4 as shown in Fig. 1. And the ER data are the reception ping signal and amplification signal, i.e. 2 and 3 as shown in Fig. 1. We can know energy decrement by ocean environment from comparison between 1 and 2 signals. After we amplify the signal 2 to be the same energy level of signal 1, and send the amplified signal 3 to sensor array. Finally we estimate the energy of 4 as target echo. This experimental result will be shown in section 4.

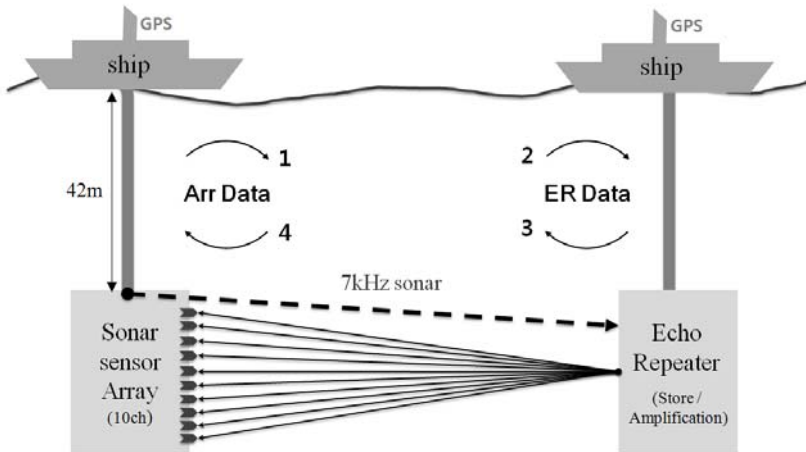


Fig. 1. Experimental data acquisition method in a real ocean: Use LFHUSS as source and 1.5m length 10ch @7kHz sonar sensor array

3 Proposed Algorithm

The proposed algorithm has two stages. The first stage is the source separation algorithm. This consists of three steps such as the principal component analysis (PCA) followed by an independent component analysis (ICA) with the fast Fourier transform (FFT). The first step is preprocessing based on PCA for removing ill-posed problem. The second step performs an ICA to estimate of the source signal from the mixtures containing noises. The last step applies the FFT and distinguishes the source signal in which we assume that the frequency property of a target echo is already known. And the second stage, after finding an original target echo, the energy estimation solution is proposed by considering an inverse procedure of the first stage. The aim of proposed algorithm is to reconstruct the target echo without noises and measure more pure energy of the reconstructed signal.

3.1 First Stage: Source Separation Algorithm

Before source separation, we use the band pass filter [11] which has 7kHz center frequency and 800Hz bandwidth as preprocessing for improving performance of source separation algorithm. This can remove noise signals which have different frequency range with the target echo.

The source separation can be obtained by optimizing an objective function which can be a scalar measure of some distribution properties of the output $y(t)$. More general measures are entropy, mutual independence, divergence between joint distribution of $y(t)$ and some given mode and higher order de-correlation. The ICA method can be formulated as optimization of a suitable objective function which is also termed as the contrast function. There are many available algorithms for ICA, but for this paper we present the results based on the Fast ICA algorithm [10].

For independent sources, we want to de-correlate the sensor signals. So, we use PCA before ICA. The purpose of PCA is to derive a relatively small number of de-correlated linear combinations of a set of random variables while retaining as much of the original information as possible. It can be used for dimensionality reduction in a data set by retaining those characteristics of the data set that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. The second order de-correlation means to make independence each other. We can solve ill-posed problem from this step.

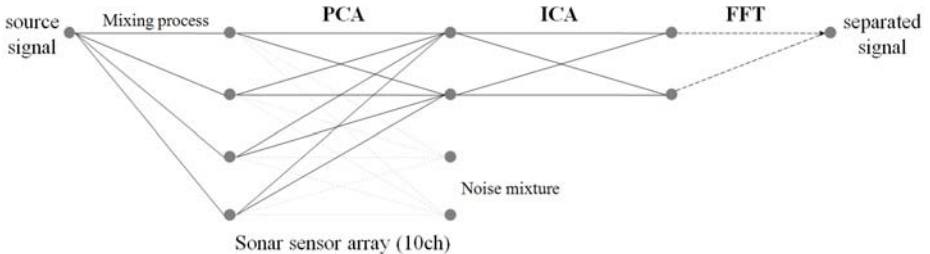


Fig. 2. The structure of source separation algorithm: This algorithm has three steps. The first step is a PCA for removing ill-posed problem. The second step is an ICA for separating signals and last step is FFT for distinguishing the source signal.

Last, we apply FFT [12] of separated signals. The FFT is an efficient algorithm to compute the discrete Fourier transform (DFT). We use this method to select one signal that has the same frequency of target echo. Finally we can obtain the estimated signal without noises for energy measurement. Figure 2 shows the overall block diagram for the source separation algorithm.

3.2 Energy Estimation Solution

The blind source separation algorithm has some constraints like amplitude and permutation indeterminacies [13]. Consequently, the energy of estimated signal is not maintained after applying first stage. Therefore we need to reconstruct the amplitude

which is changed for measuring more accurate energy. So we propose the energy estimation solution. We can solve this problem as calculating the pseudo inverse matrix of following three matrices as shown in Fig. 3.

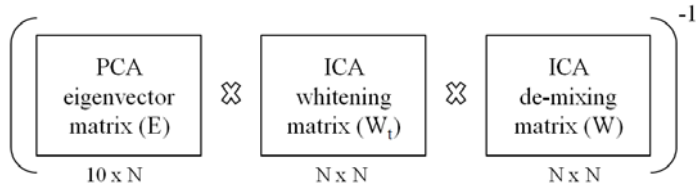


Fig. 3. Method of calculating the pseudo inverse matrix: N is the reduced dimension after PCA step of the first stage

Figure 3 shows the process of calculating the pseudo inverse matrix. N means the dimension which is reduced after PCA stage. The value of N can be controlled by setting a threshold. In reduction process of the principle components, we may lose the energy of original target echo. And then, we consider the ICA step of first stage and apply the whitening as preprocessing step. The indeterminacy associated with the ICA model is that the independent components and the columns of the mixing matrix A can be estimated up to a multiplicative constant, because any constant multiplying one independent component in the basic ICA model could be cancelled by dividing the corresponding column of the mixing matrix A by the same constant [14]. So, to make the independent components unique up to a multiplicative sign, the sources should have unity variance. This is whitening, and we must consider the whitening matrix W_t for estimating more accurate energy of target echo. And we also consider the de-mixing matrix W which is the inverse matrix of A . After calculating the pseudo inverse matrix as shown in Fig. 3, multiply that and the reconstructed signal which is estimated signal on third step of first stage. This time, remaining 9 channels signals except estimated signal put zero. After being this process, 10 channels signals are pure target echo without noises. Finally, we can reconstruct one signal of 10 channels signal average for energy measurement of target echo.

4 Experimental Results

In this section, we show some results of the proposed algorithm. We compare the result of proposed method with beam forming method [6]. In order to describe the performance of the proposed algorithm, we use the *signal-to-noise ratio* (SNR) as

$$SNR(dB) = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right). \tag{4}$$

We calculate that the signal is the range of 500Hz bandwidth at center frequency of 7kHz and the noises are the others. We obtain experiment data in real-ocean. We use the Arr data which are recorded signals at 10 channels sensor array. We synchronize each ping at sensor array and echo repeater, and estimate the target echo range. After

then we can decide the ICA input range. We apply the experiment twice without filtering and after filtering as preprocessing. In this experiment, we don't know the source signal precisely. The only way to evaluate the performance is to compare SNR of the beam forming results. Figures 4 (a) and (b) shows the separated results after filtering and the reconstructed signals by proposed method and beam forming method, respectively.

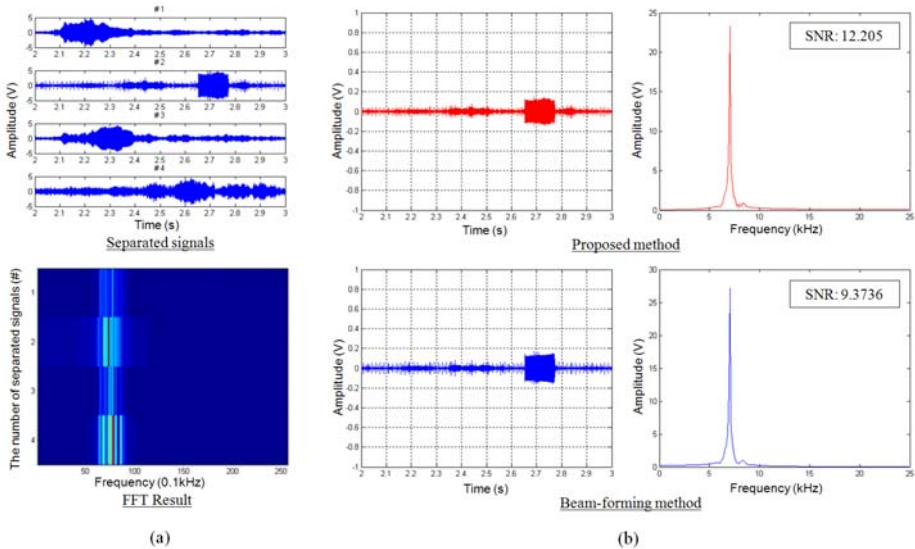


Fig. 4. The result of real-ocean recorded data after filtering: (a) Source separation and FFT result (b) Performance of the proposed method and beam forming method

Table 1 shows the SNR about 5 ping data that are recorded in the real ocean. The proposed method is the best performance at every ping. So we can efficiently reduce the noises and estimate more accurate energy of the source signal.

Table 1. The SNR of each method

Ping	without filtering		after filtering	
	beamforming method (dB)	proposed method (dB)	beamforming method (dB)	proposed method (dB)
1	-0.6584	2.7689	10.0598	12.3519
2	7.5725	10.5834	11.2264	13.7855
3	6.7827	10.1284	9.3736	12.205
4	5.8631	9.6867	10.6276	12.2378
5	-0.3869	2.2493	11.437	13.3675

5 Conclusion

We proposed echo energy estimation algorithm of active sonar in a sensor array. The source separation algorithm and the energy estimation solution have been proposed to solve pure energy estimation problem. This work describes a new application of ICA in an active sonar signal. In our experiments, our algorithm works well for the real-ocean recorded data. The developed technique not only improves SNR but also successfully distinguishes signals according to their sources of origins. In this study, we try to estimate more accurate energy level using basic signal processing algorithms. Although experimental data is not sufficiently, we approach new concept of sonar energy estimation using independent component analysis in underwater environment.

Acknowledgments. This research was supported by Korean Agency for Defense Development (ADD) under contract number UE090017DD.

References

1. Nielsen, R.O.: *Sonar Signal Processing*. Artech House Inc., Nortwood (1991)
2. Waite, D.: *Sonar for practicing Engineers*. John Wiley and Sons, New York (2003)
3. Hertz, D., Ziskind, I.: Fast approximate maximum likelihood algorithm for single source localization. *IEE Proc. Radar, Sonar, Navig.* 142(5), 232–235 (1995)
4. Porat, B., Friedlander, B.: Analysis of the asymptotic relative efficiency of the MUSIC algorithm. *IEEE Trans. Acoust., Speech, Signal processing* 4, 532–543 (1988)
5. Gavish, M., Weiss, A.J.: Performance analysis of bearing-only target location algorithms. *IEEE Trans. Aerosp. Electron. Syst.* AES-28(3), 817–827 (1992)
6. Curtis, T.E., Ward, R.J.: Digital Beamforming for Sonar. *IEE Proc., Part F, Comms., Radar and Signal Processing* 127 (August 1980)
7. Amari, S., Cardoso, J.F.: Blind source separation - semiparametric statistical approach. *IEEE Trans. Signal Processing* 45(11), 2698–2700 (1997)
8. Hyvarinen, A., Oja, E.: Independent component analysis: Algorithms and applications. *Neural Networks* 13(4-5), 411–430 (2000)
9. Cardoso, J.F.: Blind signal separation: Statistical principles. In: *Proceedings of the IEEE, October 1998, vol. 86(10)*, pp. 2009–2025 (1998)
10. Hyvrinen, A., Oja, E.: Independent component analysis: Algorithms and applications. *Neural Networks* 13(4-5), 411–430 (2000)
11. Ricardo, A.: *Practical FIR Filter Design in MATLAB*, January 12, 2004. The MathWorks, Inc., 3 Apple Hill Dr. Natick (2004)
12. FFT Tutorial, University of Rhode Island Department of Electrical and computer Engineering ELE 436, communication Systems
13. Parra, L., Spence, C.: Convolutional blind separation of non-stationary sources. *IEEE Trans. Speech and Audio Processing* 8(3) (May 2000)
14. Tong, L., Liu, R.-W., Soon, V.C., Huang, Y.-F.: Indeterminacy and identifiability of blind identification. *IEEE Trans. Circuits and Systems* 38(5) (May 1991)

Improving SVM Classification with Imbalance Data Set

Zhi-Qiang Zeng¹ and Ji Gao²

¹ Department of Computer Science and Technology, Xiamen University of Technology,
361024 Xiamen, China

² Department of Computer Science and Engineering, Zhejiang University,
310027 Hangzhou, China
lbxzzq@163.com
gaoji@mail.hz.zj.cn

Abstract. In view of inconsistent problems caused by that Synthetic Minority Over-sampling Technique (SMOTE) and Support Vector Machine (SVM) work in different space, this paper presents a kernel-based SMOTE approach to solve classification with imbalance data set by SVM. The method first preprocesses the data by oversampling the minority instances in the feature space, then the pre-images of the synthetic samples are found based on a distance relation between feature space and input space. Finally, these pre-images are appended to the original dataset to train a SVM. Experiments on real data set indicate that compared with SMOTE approach, the samples constructed by the proposed method have the higher quality. As a result, the effectiveness of classification by SVM on imbalance data set is improved.

Keywords: Imbalance, Classification, Support vector machine, Pre-image.

1 Introduction

Support vector machine (SVM) [1] is a new machine learning method which is based on the statistical learning theory developed by Vapnik et al., and it has gained wide acceptance because of the high generalization ability for a wide range of applications. Given a dataset, SVM aims at finding the discriminating hyperplane that maintains an optimal margin from the boundary examples called support vectors. An SVM, thus, focuses on improving generalization on training data. A number of recent works, however, have highlighted that the orientation of the decision boundary for an SVM trained with imbalance data, is skewed towards the minority class, and as such, the prediction accuracy of minority class is low compared to that of the majority ones (in the remainder of this paper negative is always taken to be the majority class and positive is the minority class).

A popular approach towards solving these problems is to bias the classifier so that it pays more attention to the positive instances. This can be done, for instance, by increasing the penalty associated with misclassifying the positive class relative to the negative class [2]. The net effect is that the boundary is pushed more towards the negative instances. However, a consequence of this is that SVM becomes more

sensitive to the positive instances and obtains stronger cues from the positive instances about the orientation of the plane than from the negative instances. If the positive instances are sparse, as in imbalance data set, then the boundary may not have the proper shape in the input space [3]. Another approach is to preprocess the data by undersampling the majority class or oversampling the minority class in order to balance out data set. In [4,5,6], Kubat and Matwin et al proposed an one-sided selection process which undersampled the majority class in order to create a balance data set. Though the approach does improve SVM performance, there is an inherent loss of valuable information in this process [7]. Chawla et al [8] devised a method called Synthetic Minority Oversampling Technique (SMOTE). This technique involved creating new instances through “phantomtransduction”. Experiments show that this technique to be more useful for SVM than undersampling or random oversampling. SMOTE has gained popularity in solving imbalance problem due to its performance, and researchers put forward a number of the new improved algorithms based on it. For instance, Akbani raised a method called SDC [3] which combines SMOTE with different costs, and Yang Liu proposed an approach named EnSVM [9] which connects SMOTE with boost method. The experiments show that these improved algorithms have achieved good results in imbalance data set. However, SVM works in the feature space, and SMOTE processes in the input space. This is somewhat unnatural. Because the kernel function usually implies an implicitly nonlinear mapping from the input space to the feature space, the optimal instance generated in input space is not necessarily the optimal one in feature space. Based on this, a novel SMOTE-type method called KSMOTE (Kernel-based SMOTE) is presented to overcome the problem in this paper. Different with original SMOTE, KSMOTE creates new positive instances in feature space, thereby, the inconsistency caused by processing instances in different space has been resolved.

This paper is organized as follows. Section 2 gives a brief introduction to the theoretical background with reference to classification principles of SVM. Section 3 introduces our proposed method in detail. Experimental results are demonstrated in Section 4 to illustrate the effectiveness of the proposed method. Conclusions are included in Section 5.

2 Support Vector Machine

Given training vectors $x_i \in R^h$, $i=1, \dots, l$ in two classes, and a vector $y \in R$ such that $y_i \in \{-1, 1\}$, the support vector technique requires the solution of the following optimization problem [1]:

$$\begin{aligned} \min_{w, b, \xi} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i, \\ \text{subject to} \quad & y_i (w^T \varphi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \dots, l. \end{aligned} \tag{1}$$

where ξ_i for $i=1, \dots, l$ are slack variables introduced to handle the non-separable case.

The constant $C > 0$ is the penalty parameter that controls the trade-off between the separation margin and the number of training errors, with higher value of C focusing more on minimizing error. Using the Lagrange multiplier method, one can easily obtain the following Wolfe dual form of the primal quadratic programming problem:

$$\begin{aligned} \min_{\alpha_i, i=1, \dots, l} \quad & \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j k(x_i, x_j) - \sum_{i=1}^l \alpha_i, \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C, i = 1, \dots, l, \\ & y^T \alpha = 0. \end{aligned} \tag{2}$$

SVM works in the feature space F via some nonlinear mapping function $\varphi: R^h \mapsto F$, which can be defined implicitly by a kernel function $k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$. At the optimal point for (2), either $\alpha_i = 0$ or $0 < \alpha_i < C$ or $\alpha_i = C$. The input vectors for which $\alpha_i > 0$, are termed as support vectors. These are the only important information from the perspective of classification, as they define the decision boundary, while the rest of the inputs may be ignored. For a binary classification problem, the decision function of SVM takes the form [1]:

$$f(x) = \text{sgn}\left(\sum_{i=1}^{N_s} \alpha_i k(x_i, x) + b\right), \tag{3}$$

where α_i is corresponding weight of support vector x_i , x is the input pattern to be classified, N_s is the number of support vectors and b is the bias.

3 Proposed Approach

In this section, a novel approach is presented to train SVM with imbalance data. This method consists of the following steps: First, the approach extends SMOTE algorithm to create new positive instances in the feature space F . Second, for each new instance, it finds the pre-image of the new instance in F . Finally in the third step, these pre-images of the new instances are appended to positive data set to train a SVM classifier. Experiments on real data set show the effectiveness of the proposed method in solving classification with imbalance data.

3.1 Kernel-Based SMOTE

The SMOTE approach creates new instances through “phantomtransduction”. For each positive instance, its nearest positive neighbors were identified and new positive instances were created and placed randomly in between the instance and its neighbors. This technique is more useful for SVM than undersampling or random oversampling. However, SVM works in the feature space, and SMOTE processes in the input space. This is somewhat unnatural. Because the optimal instance generated in input space is not necessarily the optimal one in feature space. To resolve the inconsistency, we have to extend the original SMOTE algorithm into feature space by using kernel methods and develop the Kernel-based SMOTE (KSMOTE) algorithm to generate

new positive instances in feature space instead of in input space. The KSMOTE algorithm is described as following:

Suppose that the positive sample set is $D^+ = \{x_1, x_2, \dots, x_n\}$, $x_i \in R^h, i=1, \dots, n$, φ is a nonlinear mapping function to project x_i into feature space F , which is associated with a kernel function $k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$.

KSMOTE(D^+ , N , k)

Input: Positive sample set D^+ ; Oversampling ratio N , namely number of synthetic samples to $|D^+|$; Number of nearest neighbors k

Output: Synthetic positive sample set D_s

Function:

getRandomPoint(S): return randomly an element from a collection S

getFeatureNeighbors(x , S , k): return k nearest neighbors of point x from collection S in the feature space

getRandomNumber($value1$, $value2$): return an arbitrary value between $value1$ and $value2$

getASCOrder(S , A): sort the collection S by ascend based on value of array A ,

getFirstPoints(S , k): return the first k elements from collection S

Algorithm:

- 1) $T := |D^+|$, $D_s := \emptyset$;
- 2) if $N < 1$ then
- 3) $T := \lfloor N \times T \rfloor$, $N := 1$;
- 4) end if
- 5) $N := \lfloor N \rfloor$, $Z := D^+$;
- 6) for $i := 1$ to T
- 7) { $x_i := \text{getRandomPoint}(Z)$;
- 8) $D_i^+ := \text{getFeatureNeighbors}(x_i, D^+ - \{x_i\}, k)$;
- 9) for $j := 1$ to N
- 10) { $x_j := \text{getRandomPoint}(D_i^+)$;
- 11) $\lambda_{ij} := \text{getRandomNumber}(0, 1)$;
- 12) $O_{ij} := \varphi(x_i) + \lambda_{ij} \times (\varphi(x_j) - \varphi(x_i))$; //create new samples in feature space (4)
- 13) $D_s := D_s \cup \{O_{ij}\}$;
- 14) $D_i^+ := D_i^+ - \{x_j\}$;
- 15) $Z := Z - \{x_i\}$;
- 16) return D_s

```

Function: getFeatureNeighbors( $x, S, k$ )
17) for  $i := 1$  to  $|S|$ 
18) {  $d_i := \|\varphi(x_i) - \varphi(x)\| = \sqrt{k(x_i, x_i) - 2k(x_i, x) + k(x, x)}$ ; // calculate
    the distance between  $x_i$  and  $x$ , where  $x_i \in S$  (5)
19)      $A[i] := d_i$ ;
20)  $S := \text{getASCOrder}(S, A)$ ;
21)  $B := \text{getFirstPoints}(S, k)$ ;
22) return  $B$ 
    
```

3.2 Solution of Pre-image Problem

It is clear that the synthetic instances in F derived from KSMOTE algorithm cannot be used directly, thus, we must use their pre-images in input space. Because the inverse map $\varphi^{-1} : F \rightarrow R^h$ is not available, it is impossible to get exact pre-image of synthetic instance by $u_{ij} = \varphi^{-1}(O_{ij})$. Thus, we need to seek an approximate solution instead. In [10], Kwok and Tsang present a method to find the approximate pre-images of patterns that are denoised in feature space via kernel principal component analysis (KPCA). We follow their strategy to seek the pre-images of synthetic samples generated by KSMOTE algorithm.

The feature-space distance between synthetic sample O_k and an arbitrary point x_i can be calculated as following:

$$\begin{aligned}
 \tilde{d}_i^2(O_{ij}, \varphi(x_i)) &= \tilde{d}_i^2(\varphi(x_i) + \lambda_{ij} \times (\varphi(x_j) - \varphi(x_i)), \varphi(x_i)) & (6) \\
 &= \|\varphi(x_i) + \lambda_{ij} \times (\varphi(x_j) - \varphi(x_i)) - \varphi(x_i)\|^2 \\
 &= (1 + 2\lambda_{ij})k(x_i, x_i) - 2k(x_i, x_i) - 2\lambda_{ij}k(x_i, x_j) + \\
 &(\lambda_{ij} - 1)^2k(x_i, x_i) + 2\lambda_{ij}(1 - \lambda_{ij})k(x_i, x_j) + \lambda_{ij}^2k(x_j, x_j) .
 \end{aligned}$$

Suppose the pre-image of O_{ij} is u_{ij} in input space, for the Gaussian kernel, the following simple relation holds true between $\tilde{d}_i^2(\varphi(u_{ij}), \varphi(x_i))$ and $d_i^2(u_{ij}, x_i)$ [11]:

$$\begin{aligned}
 \tilde{d}_i^2(\varphi(u_{ij}), \varphi(x_i)) &= \|\varphi(u_{ij}) - \varphi(x_i)\|^2 = k(u_{ij}, u_{ij}) - 2k(u_{ij}, x_i) + k(x_i, x_i) & (7) \\
 &= 2 - 2\exp(-\|u_{ij} - x_i\|^2 / (2\sigma^2)) = 2 - 2\exp(-d_i^2(u_{ij}, x_i) / (2\sigma^2)) \\
 &\Rightarrow d_i^2(u_{ij}, x_i) = -2\sigma^2 \ln(1 - \frac{1}{2}\tilde{d}_i^2(\varphi(u_{ij}), \varphi(x_i))) .
 \end{aligned}$$

Because the feature-space distance $\tilde{d}_i^2(O_{ij}, \varphi(x_i))$ is available from (6), the corresponding input-space distance between the desired approximate pre-image u_{ij} of O_{ij} and x_i can be calculated based on (7). Generally, the distances with neighbors are the most important in determining the location of any point. Hence, we will only consider the (squared) input-space distances between synthetic sample O_{ij} and its t nearest neighbors $\{\varphi(x_i^{ij}), \varphi(x_j^{ij}), \dots, \varphi(x_t^{ij})\} \subset D^+$ in F . Define a vector

$$d^2 = [d_1^2, d_2^2, \dots, d_t^2]^T , \tag{8}$$

where $d_l, l = 1, \dots, t$, are the input-space distance between the desired pre-image of O_{ij} and x_l . In [12], one attempts to find a representation of a further point with known distance from each of other points. Thus, we can use the idea to transform O_{ij} back to the input space. For the t neighbors $\{\varphi(x_1^{ij}), \varphi(x_2^{ij}), \dots, \varphi(x_t^{ij})\}$ of O_{ij} in F , we will first center them at their centroid $\bar{x} = (1/t) \sum_{l=1}^t x_l^{ij}$ and define a coordinate system in their span. First, we construct the $h \times t$ matrix $X = [x_1^{ij}, x_2^{ij}, \dots, x_t^{ij}]$ and a $t \times t$ centering matrix

$$H = I - \frac{1}{t} 11^T, \tag{9}$$

where I is a $t \times t$ identity matrix and $1 = [1, 1, \dots, 1]^T$ is a $t \times 1$ vector. The XH will center the x_l^{ij} 's at their centroid

$$XH = \begin{bmatrix} x_1^{ij} - \bar{x} & x_2^{ij} - \bar{x} & \dots & x_t^{ij} - \bar{x} \end{bmatrix}. \tag{10}$$

Suppose that XH is of rank q , we can obtain the singular value decomposition (SVD) of the $h \times t$ matrix XH as:

$$XH = [E_1, E_2] \begin{bmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = E_1 \Lambda_1 V_1^T = E_1 \Gamma, \tag{11}$$

where $E_1 = [e_1, e_2, \dots, e_q]$ is a $h \times q$ matrix with orthonormal columns e_l , and $\Gamma = \Lambda_1 V_1^T = [c_1, c_2, \dots, c_t]$ is a $q \times t$ matrix with columns c_l being the projection of $x_l^{ij} - \bar{x}$ onto the E_l . Note that, $\|c_l\|^2 = \|x_l^{ij} - \bar{x}\|^2, l = 1, \dots, t$, and collect this into a t -dimensional vector, as $d_0^2 = [\|c_1\|^2, \|c_2\|^2, \dots, \|c_t\|^2]^T$. It is clear that the location of the pre-image u_{ij} is obtained by requiring $d^2(u_{ij}, x_l^{ij}), l = 1, \dots, t$ to be as close to those values in (8) as possible, i.e.,

$$d^2(u_{ij}, x_l^{ij}) \approx d_l^2, l = 1, \dots, t. \tag{12}$$

Define $\tilde{c} \in R^{q \times 1}$ via $E_1 \tilde{c} = u_{ij} - \bar{x}$, then

$$d_l^2 \approx \|u_{ij} - x_l^{ij}\|^2 = \|(u_{ij} - \bar{x}) - (x_l^{ij} - \bar{x})\|^2 = \|\tilde{c}\|^2 + \|c_l\|^2 - 2(u_{ij} - \bar{x})^T (x_l^{ij} - \bar{x}), l = 1, \dots, t. \tag{13}$$

We sum these equations over l , and the summation of cross-product term in (13) is zero because of centering of XH . Thus

$$\sum_{l=1}^t d_l^2 = t \|\tilde{c}\|^2 + \sum_{l=1}^t \|c_l\|^2 \Rightarrow \|\tilde{c}\|^2 = \frac{1}{t} \sum_{l=1}^t (d_l^2 - \|c_l\|^2), l = 1, \dots, t, \tag{14}$$

which can be substitute for $\|\tilde{c}\|^2$ in (13), giving after a little rearrangement

$$2(x_l^{ij} - \bar{x})^T (u_{ij} - \bar{x}) = \|c_l\|^2 - d_l^2 - \frac{1}{t} \sum_{l=1}^t (\|c_l\|^2 - d_l^2), l = 1, \dots, t. \tag{15}$$

Expressing (15) in terms of matrix form, we can obtain that

$$2\Gamma^T \tilde{c} = (d_0^2 - d^2) - \frac{1}{t} 11^T (d_0^2 - d^2). \tag{16}$$

Now, $\Gamma 11^T = 0$ because of the centering. Hence, we have that

$$\tilde{c} = \frac{1}{2}(\Gamma\Gamma^T)^{-1}\Gamma(d_0^2 - d^2) = \frac{1}{2}\Lambda_1^{-1}V_1^T(d_0^2 - d^2). \quad (17)$$

Finally, by transforming \tilde{c} back to the original coordinated system in input space, the approximate pre-image of synthetic sample in F is

$$u_{ij} = \frac{1}{2}E_1\Lambda_1^{-1}V_1^T(d_0^2 - d^2) + \bar{x}. \quad (18)$$

3.3 The Main Algorithm

The sketch of our proposed method can be summarized as following:

- 1) Synthesize positive samples in the feature space F by KSMOTE method
- 2) Find the pre-images of synthetic samples in F following the approach described in section 3.2.
- 3) Regard these pre-images of synthetic instances as positive samples and append them to original data set to train a SVM classifier.

4 Experiments and Discussions

4.1 Experimental Settings

The proposed method is implemented in Matlab 7.0 and VC++ 6.0. The LIBSVM [13] is used for SVM implementation. In our experiments, we compared the performance of our classifier with regular SVM, Biased SVM and SMOTE. Six imbalance data set from UCI machine learning repository [14] are used in experiments (Table 1). Each data set was randomly split into train and test sets in the ratio 3 to 1, while sampling them in a stratified manner to ensure each of them had the same negative to positive ratio. The Gaussian kernel is used in all experiments. The parameter C and variance of Gaussian kernel are obtained by doing 10-fold cross validation on a small subset.

Table 1. Statistics of experimental datasets

Data Set	Total Instances	# of Positive Instances	# of Negative Instances	Imbalance Ratio
Abalone	4,177	32	4,145	129.53
Car	1,728	69	1,659	24.04
Glass	214	29	185	6.38
Phoneme	5,404	1,586	3,818	2.41
Pima	768	268	500	1.87
Segmentation	210	30	180	6

In order to evaluate classifiers on highly imbalance data set, using accuracy as a metric is virtually useless. This is because with an imbalance of 99 to 1, a classifier that classifies everything negative will be 99% accurate, but it will be completely useless as a classifier. The medical community, and increasingly the machine learning

community [2], use two metrics, the sensitivity and the specificity, when evaluating the performance of various tests. Sensitivity can be defined as the accuracy on the positive instances (true positives / (true positives + false negatives)), while specificity can be defined as the accuracy on the negative instances (true negatives / (true negatives + false positives)). Kubat et al [15] suggested the g-means metric defined as:

$$g = \sqrt{acc^+ \cdot acc^-} \quad (19)$$

Where acc^+ = sensitivity and acc^- = specificity. This metric has been used by several researchers for evaluating classifiers on imbalance data set [3, 6, 7]. We will also use this metric to evaluate our classifier.

4.2 Results and Discussions

Table 2 shows the performance of the four algorithms using g-means metric. For the SMOTE and KSMOTE, numbers included in bracket denote corresponding oversampling rate. The last line of the table is the arithmetic mean of each algorithm over all the g-means metrics. This arithmetic mean can be used to quantify the overall performance of each algorithm over all six data set.

Table 2. Performance of the four algorithms using g-means metric

Data Set	SVM	Biased SVM	SMOTE (<i>N</i>)	KSMOTE (<i>N</i>)
Abalone	0	0.8137	0 (10)	0 (10)
Car	0	0.3227	0.9884 (5)	0.9875 (5)
Glass	0.8658	0.8814	0.9236 (1)	0.9328 (1)
Phoneme	0.8276	0.8312	0.8347 (1)	0.8543 (1)
Pima	0.7119	0.7326	0.7456 (1)	0.7833 (1)
Segmentation	0.9184	0.9366	0.9773 (1)	0.9865 (1)
Mean	0.5540	0.7530	0.7449	0.7574

As shown in Table 2, mean value of SVM is the smallest among four algorithms, which indicates that performance of regular SVM on imbalance data set is poor. The mean value of Biased SVM is litter bigger than SMOTE. It is due to the fact that g-means of SMOTE is zero on, highly imbalance data set, Abalone, which significantly degraded its corresponding mean value. Overall, apart from Abalone, g-means of SMOTE on five other data sets is much bigger than that of Biased SVM. It indicates that SMOTE algorithms are superior to Biased SVM in whole. It also can be seen that mean value of KSMOTE is the biggest among all algorithms. On most of the data set, g-means of KSMOTE is bigger than that of SMOTE, which indicates that KSMOTE algorithm outperforms SMOTE algorithm under current oversampling rate.

We also investigated the effect of varying oversampling rate on SMOTE and KSMOTE. Fig.1 presents the values of g-mean on three data set, corresponding to each oversampling rate. Results show that KSMOTE outperforms SMOTE over the whole range of oversampling rate.

From Table 2 and Fig 1, it can be concluded that quality of synthetic instances generated by KSMOTE is superior to that created by SMOTE, consequently, KSMOTE algorithm is more efficient in solving classification with imbalance data set.

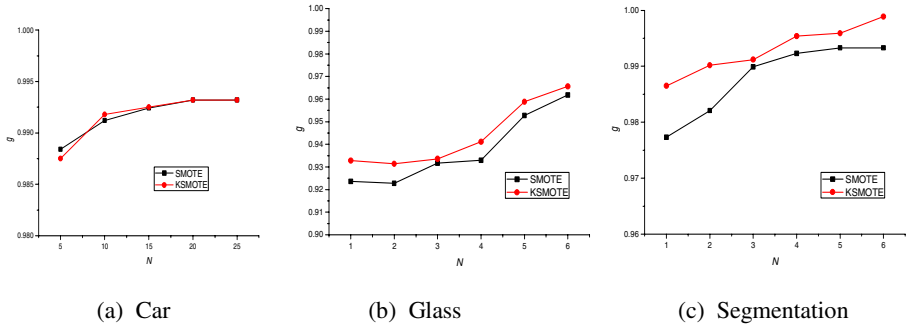


Fig. 1. G-mean for change of oversampling rate on Car, Glass and Segmentation datasets

5 Conclusions

In this paper, a new strategy is presented to improve prediction accuracy of SVM for imbalance data. The proposed strategy constructs minority instances in the feature space to balance out dataset, which yields better recognition performance for imbalance data. In contrast to existing schemes like SMOTE which generates minority instances in the input space, the samples constructed by the proposed method have the higher quality. Experiments on real data set indicate that the effectiveness of classification by SVM on imbalance data set is improved.

References

1. Vapnik, V.: *Statistical Learning Theory*. Wiley, New York (1998)
2. Veropoulos, K., Campbell, C., Cristianini, N.: Controlling the sensitivity of support vector machines. In: *Proceedings of the International Joint Conference on AI*, pp. 55–60 (1999)
3. Akbani, R., Kwek, S., Japkowicz, N.: Applying Support Vector Machines to Imbalance data set. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) *ECML 2004*. LNCS (LNAI), vol. 3201, pp. 39–50. Springer, Heidelberg (2004)
4. Yuan, J., Li, J., Zhang, B.: Learning concepts from large scale imbalanced data sets using support cluster machines. In: *Proc. of the ACM Int'l Conf. on Multimedia*, pp. 441–450 (2006)
5. Kang, P., Cho, S.: EUS SVMs: Ensemble of Under-Sampled SVMs for Data Imbalance Problems. In: King, I., Wang, J., Chan, L.-W., Wang, D. (eds.) *ICONIP 2006*. LNCS, vol. 4232, pp. 837–846. Springer, Heidelberg (2006)
6. Li, P., Wang, X., Liu, Y., Wang, X.: A Classification Method for Imbalance Data Set Based on Hybrid Strategy. *Chinese Journal of Electronics* 35(11), 2161–2165 (2007)
7. Imam, T., Ting, K.M., Kamruzzaman, J.: z-SVM: An SVM for improved classification of imbalanced data. In: Sattar, A., Kang, B.-h. (eds.) *AI 2006*. LNCS (LNAI), vol. 4304, pp. 264–273. Springer, Heidelberg (2006)
8. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)* 16, 321–357 (2002)

9. Liu, Y., An, A., Huang, X.: Boosting Prediction Accuracy on Imbalance data set with SVM Ensembles. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) PAKDD 2006. LNCS (LNAI), vol. 3918, pp. 107–118. Springer, Heidelberg (2006)
10. Kwok, J.T., Tsang, I.W.: The pre-image problem in kernel methods. *IEEE Transactions on Neural Networks* 15(6), 1517–1525 (2004)
11. Williams, C.K.I.: On a connection between kernel PCA and metric multidimensional scaling. *Machine Learning* 46(1/3), 11–19 (2002)
12. Gower, J.C.: Adding a point to vector diagrams in multivariate analysis. *Biometrika* 55(3), 582–585 (1968)
13. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
14. Murphy, P.M., Aha, D.W.: UCI repository of machine learning databases, Irvine, CA (1994), <http://www.ics.uci.edu/~mlearn/MLRepository.html>
15. Kubat, M., Matwin, S.: Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. In: *Proceedings of the 14th International Conference on Machine Learning* (1997)

Framework for Object Tracking with Support Vector Machines, Structural Tensor and the Mean Shift Method

Bogusław Cyganek

AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
cyganek@uci.agh.edu.pl

Abstract. In this paper a system is presented for object tracking based on the novel connection of the one-class SVM classifier with the mean shift tracker. An object for tracking is defined by feature vectors composed of the components of the orthogonal color space, as well as local phase and coherence components of the structural tensor which convey information on texture. The binary output of the SVM is mapped into a membership field with a proposed transformation function. Tracking is performed with the continuously adaptive mean shift method operating in the membership field. The method shows high discriminative power and fast run-time properties.

1 Introduction

Many computer vision systems rely on object tracking. This helps in detection and following of selected objects in a video stream. Such systems are used to track faces, road signs, cars, pedestrians, or any other object with sufficiently discriminative features.

The proposed tracking system relies on the Support Vector Machine (SVM) classifier operating in the one-class mode (OC-SVM) [10]. This is rather rare mode of operation of SVM. This shows useful in some situations in which number of points defining an object is much lower than a number of all other points, which usually are also not known. OC-SVM was proposed and tested in a number of classification problems by Tax *et al.* [10]. OC-SVM was proposed by Cyganek in [4] to segment road signs. This type of classifier was also used by Jin *et al.* to face detection [6].

In the proposed tracking system OC-SVM is trained with features defining an object to be tracked. The feature vector is proposed to be built from the components of the orthogonal IJK color space, as well as from the components of the structural tensor (ST). The latter convey information on local structure around each pixel in an image. Tracking is done with the continuous adaptive version of the mean shift method. However, the method does not operate directly with the output of the SVM. Instead, its output is mapped to create a smooth membership field. This is obtained with the proposed transformation function.

The paper is organized as follows. Architecture of the proposed system is described in section (2). Structure of the OC-SVM used in our system is discussed in (3). Feature extraction and their postprocessing is dealt with in section (4). Then the

mean shift tracking method is outlined in (5). The paper ends with experimental results (6) and conclusions (7).

2 Architecture of the Tracking System

Fig. 1 depicts architecture of the proposed system for object tracking. The input consists of the color video stream in which an object for tracking is selected by outlining with a rectangle.

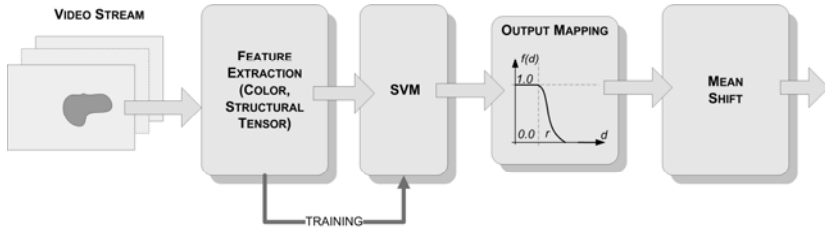


Fig. 1. Architecture of the tracking system based on the one-class SVM

Then, features are extracted from thus defined region. These are proposed to be values of the orthogonal IJK space with the components of the ST, as discussed in the next section. The features are used to train the SVM classifier. In our system it operates in the one-class mode which is suitable for relatively small objects in the large background field. However, an output of the SVM is a binary value which indicates whether a pixel belongs to an object or to the background. Usually this is too restrictive for the consecutive mean shift tracking. Therefore output of the SVM needs to be softened. This is done with the proposed mapping function. Finally, the continuous adaptive version of the mean shift is used for tracking.

3 Data Classification with Support Vector Machines

Support Vector Machines (SVM) were proposed by Vapnik [11] for binary classification. The most characteristic is transformation of data into so called feature space, in which the classification can be done with linear hypersurface. However, the new space is usually of higher dimension than the original one. The transformation is done with a kernel, frequently selected based on a type of data. In this realization, however, we propose to use the one-class SVM [10] in which the classifier is trained to recognize objects belonging only to one class, i.e. which features fall into the special hypersphere, depicted in Fig. 2.

The hypersphere is entirely characterized by its centre \mathbf{a} and a radius r . At the same time the volume of that sphere should be minimal to tightly encompass the class of interest. This volume is proportional to r^n . Nevertheless minimization of r^n means also minimization with respect to r^2 which simplifies further discussion. Hence, the minimization functional Θ is as follows

$$\Theta(\mathbf{a}, r) = r^2 \tag{1}$$

with the constraint

$$\forall_i : \|\mathbf{x}_i - \mathbf{a}\| \leq r^2, \tag{2}$$

where \mathbf{x}_i are data points.

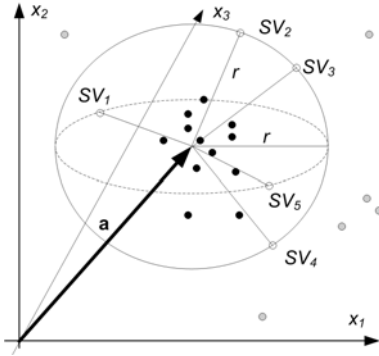


Fig. 2. The hypersphere enclosing inliers (black dots). Support vectors (SV) are on the border, outliers (gray) are outside.

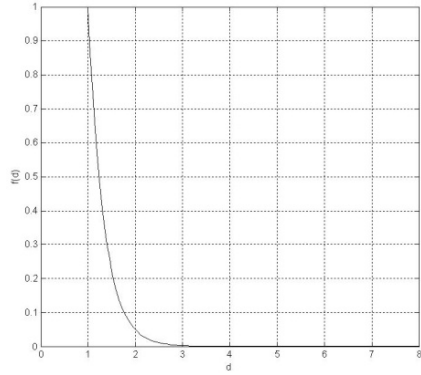


Fig. 3. Mapping of the distance d to the hypersphere into the object membership function $f(d)$

However, to introduce a possibility of some outliers in the training set further distances that r can be allowed but with some additional penalty. To accomplish this the so called slack variables ξ_i are introduced, as proposed by Vapnik [11]. This yields

$$\Theta(\mathbf{a}, r) = r^2 + C \sum_i \xi_i \tag{3}$$

with the constraints that almost all objects are within the sphere, i.e.

$$\forall_i : \|\mathbf{x}_i - \mathbf{a}\| \leq r^2 + \xi_i, \quad \xi_i \geq 0, \tag{4}$$

where ξ_i are slack variables for *each* input data x_i , and C is a parameter that controls the optimization process. The larger C , the less outliers are possible at the larger volume of the hypersphere. The summation in the above spans all N input data. Given a set of training points $\{\mathbf{x}_i\}$, solution to the equation (3) and (4) can be obtained with the Lagrange multipliers. From this a distance d from the centre \mathbf{a} of the hypersphere to a test point \mathbf{x}_x can be computed. Then if $d \leq r$, i.e.

$$d^2(\mathbf{x}_x, \mathbf{a}) \leq r^2, \tag{5}$$

then a point \mathbf{x}_x is classified as belonging to the class enclosed by this hypersphere. Otherwise, it is an outlier. It can be shown that a center \mathbf{a} of the hypersphere is [10].

$$\mathbf{a} = \sum_k \alpha_k \mathbf{x}_k, \tag{6}$$

where α_k is an (unknown) Lagrange multiplier associated with a data \mathbf{x}_k . For a distance d of the test point \mathbf{x}_x to the center \mathbf{a} of the hypersphere the following holds

$$d^2(\mathbf{x}_x, \mathbf{a}) = \|\mathbf{x}_x - \mathbf{a}\|^2 = K(\mathbf{x}_x, \mathbf{x}_x) - 2K(\mathbf{x}_x, \mathbf{a}) + K(\mathbf{a}, \mathbf{a}), \quad (7)$$

which after entering (6) leads to [4]

$$d^2(\mathbf{x}_x, \mathbf{a}) = K(\mathbf{x}_x, \mathbf{x}_x) - 2 \sum_{i \in \text{Idx}(SV)} \alpha_i K(\mathbf{x}_x, \mathbf{x}_i) + \sum_{j \in \text{Idx}(SV)} \alpha_j \sum_{i \in \text{Idx}(SV)} \alpha_i K(\mathbf{x}_i, \mathbf{x}_j), \quad (8)$$

where $K: \mathfrak{R}^n \times \mathfrak{R}^n \rightarrow \mathfrak{R}$ is a kernel function [9][11]. This can be an inner product of vectors or some other nonlinear function, such as a polynomial or a radial base function as in (11). $\text{Idx}(SV)$ denotes a set of indices of all the support vectors found for this problem. The summation in the above takes on only such \mathbf{x}_i which are support vectors (SVs), because for the inliers it holds that $\alpha_i=0$, whereas border support vectors do not fulfill the optimization criteria. The third term in (8) does not depend on the test point \mathbf{x}_x and therefore it can be precomputed. SVs are placed on the boundary of the hypersphere and thus are equidistant to its center. Therefore the following holds

$$\begin{aligned} \forall_{\mathbf{x}_s \in SV} r^2 = d^2(\mathbf{x}_s, \mathbf{a}) = \\ K(\mathbf{x}_s, \mathbf{x}_s) - 2 \sum_{i \in \text{Idx}(SV)} \alpha_i K(\mathbf{x}_s, \mathbf{x}_i) + \sum_{j \in \text{Idx}(SV)} \alpha_j \sum_{i \in \text{Idx}(SV)} \alpha_i K(\mathbf{x}_i, \mathbf{x}_j), \end{aligned} \quad (9)$$

where \mathbf{x}_s is one of the support vectors, SV denotes a set of all found support vectors for this classification task, i.e. vectors that comply with (4). This yields

$$K(\mathbf{x}_x, \mathbf{x}_x) - 2 \sum_{i \in \text{Idx}(SV)} \alpha_i K(\mathbf{x}_x, \mathbf{x}_i) \leq K(\mathbf{x}_s, \mathbf{x}_s) - 2 \sum_{i \in \text{Idx}(SV)} \alpha_i K(\mathbf{x}_s, \mathbf{x}_i), \quad (10)$$

which for the RBF kernel

$$K_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}, \quad (11)$$

simplifies to

$$\sum_{i \in \text{Idx}(SV)} \alpha_i K(\mathbf{x}_x, \mathbf{x}_i) \geq \sum_{i \in \text{Idx}(SV)} \alpha_i K(\mathbf{x}_s, \mathbf{x}_i) = \delta. \quad (12)$$

The right side of the above formula is constant in the recognition stage, thus it can be precomputed to a value δ which denotes a cumulative kernel-distance of a SV to all other SVs. Equation (12) is used to test a pattern \mathbf{x}_x if it belongs to a class represented by a set of SVs.

An alternative but equivalent formulation of the OC-SVM was stated by Schölkopf *et al.* [8]. In this formulation, instead of a hypersphere, a hyperplane is searched which maximally separates data from the origin of the coordinate system. Also, in (3) instead of the parameter C the parameter ν is used. These are related by $C=(\nu N)^{-1}$. Usually for tracking C is chosen close to 1 to indicate that all pixels defining an object are inliers. In other words in a definition of an object we do not expect outliers.

Parameter γ in (11) controls the so called spread of the kernel. The larger this parameter, the higher adaptation of the model to the training data which results in larger

number of support vectors and finally in lower generalization properties. In our system parameter γ is chosen based on some experiments and its useful range is 0.2 up to 24. A default value in our software framework is 2.

4 Feature Extraction and Preprocessing

An area of an object to be tracked is outlined with a rectangle. This can be done by a user, or can be sent from other module of the system. Then features from this region-of-interest are collected and processed for further tracking. The most natural are the RGB color components. However, during experiments other spaces showed to provide more descriptive information about regions-of-interests. This is for instance the *orthogonal IJK* space derived from the RGB proposed by Pătrașcu [7], in which

$$I = (R + G + B) / \sqrt{3}, \quad J = (2R - G - B) / \sqrt{6}, \quad \text{and} \quad K = (G - B) / \sqrt{2}. \tag{13}$$

Values of IJK are in the range of 0 to 255. However, in some cases good results are obtained if these are quantized.

Nevertheless, in many real situations color information is not sufficient to discriminate an object from its background. Therefore in many systems texture and other image features are employed. However, if these are too specific to certain regions of an object to be tracked, then tracking can be lost. Therefore a novel proposition is to use quantized components of the following vector

$$\mathbf{s} = \left[T_{xx} + T_{yy}, \quad \text{ATAN2}(2T_{xy}, T_{xx} - T_{yy}), \quad \frac{(T_{xx} - T_{yy})^2 + 4T_{xy}^2}{(T_{xx} + T_{yy})^2} \right]^T, \tag{14}$$

where T_{ij} are components of the ST. In the scale-space these can be computed as [5]

$$T_{ij}(\rho, \xi) = F_\rho(R_i^{(\xi)} R_j^{(\xi)}), \tag{15}$$

where $R_i^{(\xi)}$ is a ξ -tap discrete differentiating operator, F_ρ is a Gaussian smoothing kernel with scale ρ . In our experiments $R_i^{(\xi)}$ was a 5x5 optimized derivative filter and F_ρ was a 5x5 Gaussian smoothing mask [5]. Values of \mathbf{s} were quantized into 32-128 bins. Good results were obtained omitting the s_1 component in (14). Finally, a feature vector is created which consists of the chosen color and ST components. This is used to train the OC-SVM, as well as in the run to check if a pixel belongs to an object.

However, a missing link is to map the output of the SVM to the mean shift, since in its basic version a probability field is assumed in which a gradient of the *pdf* is traversed. It was shown that the mean shift can overcome the probability constraint and can also operate within the fuzzy membership function with values increasing for a tracked object [5]. However, output of the SVM does not follow either of the mentioned fields. In the worst case an output from the OC-SVM can be used which is ‘1’ for the object and ‘0’ for its background. However, this leads to the very sparse field

which in some cases can even impeditment tracking. Therefore a transformation function needs to be designed which maps the SVM output into a field acceptable by the mean shift. For this purpose we propose the following function

$$f(d) = \min \left\{ 1, \frac{1}{e^{c(d-r)}} \right\}, \tag{16}$$

where c is a parameter that controls falling rate, d and r are in (8) and (9). Plot of $f(d)$ for $c=3$ is depicted in Fig. 3. Thus, all points that fall inside the hypersphere in Fig. 2 are attributed 1, whereas all outliers obtain value falling exponentially toward 0 with a rate controlled by c . In our experiments good results were obtained with $3 \leq c \leq 15$.

5 Object Tracking with the Mean Shift Method

In the mean shift method gradient of the *pdf* is traced [3], i.e.

$$\nabla P(\mathbf{x}) = \frac{1}{A} \sum_{l=1}^A \frac{1}{h_l^N} \nabla k \left(\left\| \frac{\mathbf{x} - \mathbf{x}_l}{h_l} \right\|^2 \right). \tag{17}$$

where k is a kernel of non-parametric estimation of the *pdf*, h_l is a width of the kernel. Assuming L_2 Euclidean distance between vectors the above expands as follows

$$\begin{aligned} \nabla P(\mathbf{x}) &= \frac{1}{A} \sum_{l=1}^A \frac{1}{h_l^N} \nabla k \left(\frac{(\mathbf{x} - \mathbf{x}_l)^T (\mathbf{x} - \mathbf{x}_l)}{h_l} \right) = \\ &= \frac{1}{A} \sum_{l=1}^A \frac{2(\mathbf{x}_l - \mathbf{x})}{h_l^{N+2}} g \left(\frac{\|\mathbf{x} - \mathbf{x}_l\|^2}{h_l^2} \right) \end{aligned} \tag{18}$$

where

$$g(x) = -k'(x), \tag{19}$$

is a derivative of the profile of a kernel used to assess the *pdf*. It is worth noticing that usually it is a different kernel than the one used to train the OC-SVM discussed in the previous section. For instance for the popular Epanechnikov kernel K_E [3] we have the following derivative of its profile

$$g_E(x) = \begin{cases} 1, & \text{for } x < 1 \\ 0, & \text{otherwise} \end{cases}. \tag{20}$$

The steepest ascent optimization employed by the mean shift method follows direction toward a stationary point. It is a point for which $\nabla P(\mathbf{x})$ tends toward 0. This way an extreme value of $P(\mathbf{x})$ is reached at a point \mathbf{x}_m given by the following formula

$$\mathbf{x}_m = \frac{\sum_{l=1}^A \frac{\mathbf{x}_l}{h_l^{N+2}} g\left(\frac{\|\mathbf{x} - \mathbf{x}_l\|^2}{h_l^2}\right)}{\sum_{l=1}^A \frac{1}{h_l^{N+2}} g\left(\frac{\|\mathbf{x} - \mathbf{x}_l\|^2}{h_l^2}\right)}. \tag{21}$$

From the above we easily notice that \mathbf{x}_m actually is a weighted mean vector of the data samples with the weights being computed from the function g .

A variant of the above, named a continuously adaptive mean shift – a *CamShift*, was proposed by Bradski basically for face tracking in video [1]. It differs from the basic formulation of the mean shift method firstly by assumption of a square kernel K in (17). It also assumes re-computation of densities in each frame (or a tracked part of it). Assuming the rectangular Epanechnikov kernel and the model-target concordance probability provided with a function W defined for each location \mathbf{x}_l , allows reformulation of the mean shift (21), with \mathbf{x}_m formulated as follows

$$\mathbf{m} = \frac{\sum_{l=1}^A \mathbf{x}_l W(\mathbf{x}_l)}{\underbrace{\sum_{l=1}^A W(\mathbf{x}_l)}_{\mathbf{x}_m}} - \mathbf{x}. \tag{22}$$

Thus, \mathbf{x}_m is simply a centroid of the “mass” expressed by the membership field W . Actually W can be any other nonnegative signal, as alluded to previously. Hence, (22) can be rewritten in terms of the statistical central moments m_{10} , m_{01} , and m_{00} , as follows

$$\mathbf{x}_m = \begin{bmatrix} m_{10} & m_{01} \\ m_{00} & m_{00} \end{bmatrix}^T, \text{ assuming that } m_{00} \neq 0. \tag{23}$$

(23) was used in our experiments. Details of the algorithm can be found in [1].

6 Experimental Results

Computations were done in a software framework which is based on the HIL library, developed by the author and described in the book by Cyganek *et al.* [5]. Experiments were performed on the IBM PC with Pentium IV 3.4GHz and 2GB RAM. For training a modified version of the LIBSVM library was used [2]. Software allows choice of the features used for tracking (color space, color channels, structural tensor components), as well as parameters for training of the OC-SVM which in practice is the spread value γ in (11) since the parameter C is usually set close to 1, indicating that all pixels should be equally important when defining an object to be tracked. The software platform allows outline of the object with a rectangle. All pixels from this rectangle are taken for training of the OC-SVM. The training process is controlled by measuring a ratio of a number of the found support vectors to the total number of

pixels which define an object. Heuristically it was found that if this ratio is close to 0.1 then there is a compromise between accuracy and generalization properties of the tracker. After a number of frames the tracker needs to be retrained with a new appearance of the tracked object.

Fig. 4 depicts tracking of a warning road sign in a traffic sequence. The first row of Fig. 4 contains consecutive frames taken from the moving car. The middle row shows the confidence maps obtained from the trained OC-SVM exclusively with the IJK color components. Finally, the last row shows the confidence maps from the OC-SVM trained with IJK color and structural tensor components. It is visible that adding ST increases discriminative properties of the tracker. However, this is at a cost of a number of support vectors, since in the middle row it was 32 whereas in the lower one its number was 51. The larger this number, the longer execution time.

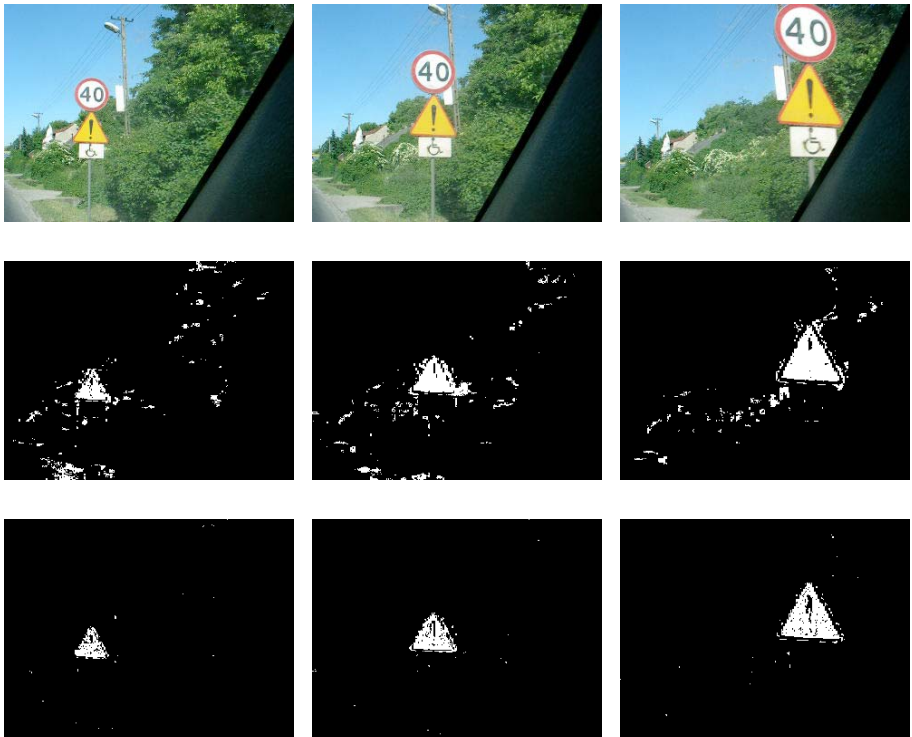


Fig. 4. Comparison of sign tracking in a traffic sequence. Upper row shows three frames. Middle row shows confidence maps for the IJK color features only. Lower row shows confidence maps for the IJK and ST features.

Fig. 5 shows an example of a car tracking in a traffic sequence (white car in the left image). A confidence map with the RGB color components is shown in the middle of Fig. 5. The right image in Fig. 5 depicts a confidence map with the IJK and ST components which together show the most discriminative properties.



Fig. 5. Car tracking example in a traffic sequence (left). A confidence map with the RGB color components (middle). A confidence map with the IJK and ST components.

Fig. 6 shows another example of tracking of a red car in a traffic sequence (top row) with the IJK and ST features. Confidence maps obtained with the OC-SVM are in the lower row. Parameter γ in this experiments was automatically found to be 1.8 using the mentioned method of measuring a ratio of support vectors to the data points.

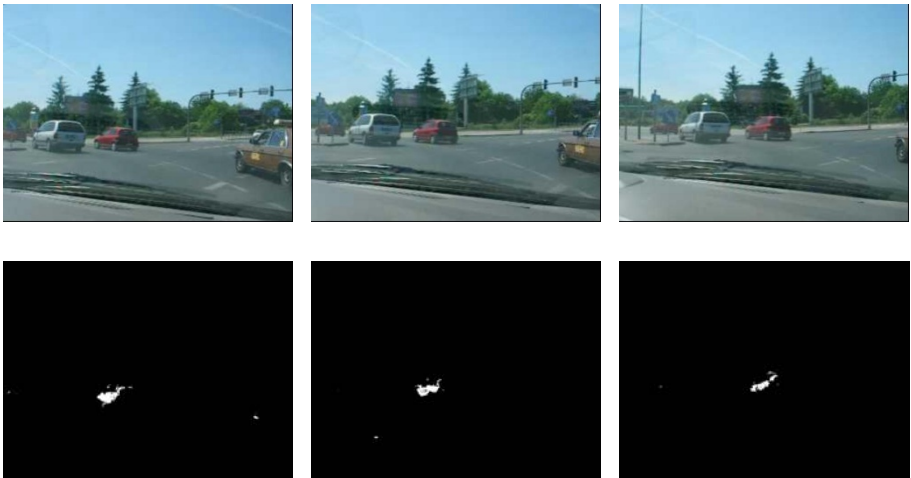


Fig. 6. Tracking of the red car in a traffic sequence (top row) with the IJK and ST features. Confidence map obtained with the OC-SVM (lower row).

It is interesting to observe that in the proposed system we employ three levels of control of the ratio of pixels belonging to an object and to the background. The first level of control are slack variables (4) of the SVM. The second control constitutes quantization values of the IJK and ST components. The last level of control is the parameter c in (16), which in the shown experiments was set to 10.

7 Conclusions

In the paper a system for object tracking is proposed. Its central part is the SVM classifier operating in the one-class mode. This is trained with the features of the object to

be tracked. Based on many experiments the five element feature vector was found to be the best trade-off between accuracy and speed. These are composed from the quantized components of the orthogonal IJK color space and two components of the structural tensor, i.e. the local orientation and the coherence factors. The binary output of the SVM is mapped by the proposed function into the membership field. This, in turn, is directly used to track objects with the continuous adaptive mean shift method.

Execution of the method allows processing of few 320x240 frames per second in our software framework, depending on the number of support vectors necessary to model an object. In our experiments this varied from about ten up to few hundred, depending on the training parameters. The method is quite easy for control since in practice one needs to choose only a spread parameter of the kernel function. This, however, can be set automatically by controlling a ratio of a number of support vectors to the amount of pixels which define an object to be tracked.

Acknowledgements

This work was supported from the Polish funds for scientific research in 2009.

References

1. Bradski, G.R.: Computer Vision Face Tracking for Use in a Perceptual User Interface, Intel (1998)
2. Chang, C.-C., Lin, C.-J.: LIBSVM, a library for support vector machines (2001), <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
3. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis And Machine Intelligence* 24(5), 603–619 (2002)
4. Cyganek, B.: Color Image Segmentation With Support Vector Machines: Applications To Road Signs Detection. *International Journal of Neural Systems* 18(4), 339–345 (2008)
5. Cyganek, B., Siebert, J.P.: *An Introduction to 3D Computer Vision Techniques and Algorithms*. Wiley, Chichester (2009)
6. Jin, H., Liu, Q., Lu, H., Tong, X.: Face Detection Using One-Class SVM in Color Images. In: 7th International Conf. on Signal Processing ICSP 2004, pp. 1431–1434 (2004)
7. Pătrașcu, V.: Fuzzy Image Segmentation Based on Triangular Function and Its n-dimensional Extension, *Fuzziness and Soft Computing*, pp. 187–207. Springer, Heidelberg (2007)
8. Schölkopf, B., Smola, A.J.: *Learning with Kernels*. MIT Press, Cambridge (2002)
9. Shawe-Taylor, J., Cristianini, N.: *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge (2004)
10. Tax, D.M.J., Duin, R.P.W.: Support Vector Data Description. *Machine Learning* 54, 45–66 (2004)
11. Vapnik, V.N.: *The Nature of Statistical Learning Theory*. Springer, Heidelberg (2000)
12. Zou, A.-M., Hou, Z.-G., Tan, M.: Support Vector Machines (SVM) for Color Image Segmentation with Applications to Mobile Robot Localization Problem. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) *ICIC 2005*. LNCS, vol. 3645, pp. 443–452. Springer, Heidelberg (2005)

Suitable ICA Algorithm for Extracting Saccade-Related EEG Signals

Arao Funase^{1,2}, Motoaki Mouri^{1,2}, Andrzej Cichocki², and Ichi Takumi¹

¹ Graduate School of Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, 466-8555, Japan

² Brain Science Institute, RIKEN, 2-1, Hirosawa, Wako, 351-0198, Japan

Abstract. Our goal is to develop a novel BCI based on an eye movements system employing EEG signals on-line. Most of the analysis on EEG signals has been performed using ensemble averaging approaches. However, it is suitable to analyze raw EEG signals in signal processing methods for BCI.

In order to process raw EEG signals, we used independent component analysis (ICA). However, we do not know which ICA algorithms have good performance. It is important to check which ICA algorithms have good performance to develop BCIs. Previous paper presented extraction rate of saccade-related EEG signals by five ICA algorithms and eight window size.

However, three ICA algorithms, the FastICA, the NG-FICA and the JADE algorithms, are based on 4th order statistic and AMUSE algorithm has an improved algorithm named SOBI. Therefore, we must re-select ICA algorithms.

In this paper, we add new algorithms; the SOBI and the MILCA. The SOBI is an improved algorithm based on the AMUSE and uses at least two covariance matrices at different time steps. The MILCA uses the independency based on mutual information. Using the Fast ICA, the JADE, the AMUSE, the SOBI, and the MILCA, we extract saccade-related EEG signals and check extracting rates.

Secondly, in order to get more robustness against EOG noise, we use improved FastICA with reference signals and check extracting rates.

1 Introduction

Brain-computer interfaces (BCIs) have been the subject of research efforts for a few decades. The capabilities of BCIs allow them to be used in situations unsuitable for the conventional interfaces. BCIs are used to connect a user and a computer via an electroencephalogram (EEG).

EEG related to fast eye movements (saccade) have been studied by our group toward developing a BCI eye-tracking system based on saccade-related EEG [1]. In previous research, EEG data were analyzed using the ensemble averaging method. Ensemble averaging is not suitable for analyzing raw EEG data because the method needs many repetitive trials.

Recording EEG data repetitively is a critical problem to develop BCIs. It is essential to overcome this problem in order to realize practical use of BCIs for single trial EEG data.

Recently, the independent component analysis (ICA) method has been introduced in the field of bio-signal processing as a promising technique for separating independent sources. The ICA method can process raw EEG data and find features related to various one's activity. Therefore, the ICA algorithm overcomes the problems associated with ensemble averaging, and the ICA analyzes the waveforms of the EEG data.

There are many algorithms to compute ICA [2]. It is important to check which ICA algorithms have good performance of analysis on EEG signals. Researchers check which ICA algorithms have good performance of extracting P300 and EOG artifact. We would like not to extract P300 signals and EOG artifact but to extract saccade-related EEG signals. Therefore, we must check which ICA algorithms can extract saccade-related EEG signal effectively.

In previous studies [3], we used the FastICA [4], the NG-FICA [5], the AMUSE [6], the JADE [7] to analyze saccade-related EEG signal. However, we must re-select an ICA algorithm since three ICA algorithms: the FastICA, the NG-FICA and the JADE algorithms are based on the 4th order statistic and the AMUSE algorithm has an improved algorithm named SOBI [8].

In this research, we add new algorithms: the SOBI and the MILCA [9]. The SOBI is an improved algorithm based on the AMUSE and uses at least two covariance matrices at different time steps. The MILCA uses the independency based on mutual information. Using the Fast ICA, the JADE, the AMUSE, the SOBI, and the MILCA, we extract saccade-related EEG signals and check extracting rates.

Secondly, we focus on window sizes of EEG signals to be analyzed. In order to analyze EEG signals in on-line system, we must choose an appropriate window size to extract continuous EEG signals. In this paper, we separate window sizes into two groups: the windows excluding EEG signals after eye movements and the windows include EEG signals after eye movements.

2 Independent Component Analysis (ICA)

The ICA method is based on the following principles (Fig. 1). Assuming that the original (or source) signals have been linearly mixed, and that these mixed signals are available, ICA recognizes in a blind manner a linear combination of the mixed signals, and recovers the original source signals, possibly re-scaled and randomly arranged in the outputs.

The $\mathbf{s} = [s_1, s_2, \dots, s_n]^T$ means n independent signals from mutual EEG sources in the brain, for example. The mixed signals \mathbf{x} are thus given by $\mathbf{x} = \mathbf{A}\mathbf{s}$, where \mathbf{A} is an $n \times n$ invertible matrix. \mathbf{A} is the matrix for mixing independent signals. In the ICA method, only \mathbf{x} is observed. The value for \mathbf{s} is calculated by $\mathbf{s} = \mathbf{W}\mathbf{x}$ ($\mathbf{W} = \mathbf{A}^{-1}$). However, it is impossible to calculate \mathbf{A}^{-1} algebraically

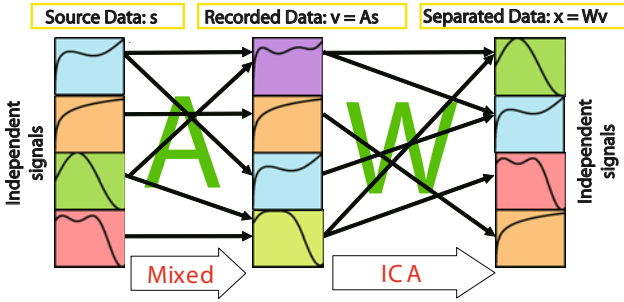


Fig. 1. Conceptual ICA algorithms

because information for A and s are not already known. Therefore, in the ICA algorithm, W is estimated non-algebraically. The assumption of the ICA algorithm is that s is mutually independent. In order to calculate W , different cost functions are used in the literature, usually involving a non-linearity that shapes the probability density function of the source signals.

3 Experimental Settings

There were two tasks in this study. The first task was to record the EEG signals during a saccade to a visual target that is either his/her right side or left side. The second task was to record the EEG signals as a control condition when a subject did not perform a saccade even though a stimulus has been displayed. First task and second task were called visual experiments. Each experiment was comprised of 50 trials in total: 25 on the right side and 25 on the left side.

The EEG signals were recorded through 19 electrodes (Ag-AgCl), which were placed on the subject's head in accord with the international 10-20 electrode position system. The Electrooculogram (EOG) signals were simultaneously recorded through two pairs of electrodes (Ag-AgCl) attached to the top-bottom side and right-left side of the right eye.

Recorded EEG signals were calculated by five ICA algorithms: FastICA, AMUSE, JADE, SOBI, MILCA. In order to calculate independent components, we must decide the window length. In this paper, there were 8 size windows.

1. Window A: -999[ms] to 1000[ms]
2. Window B: -499[ms] to 500[ms]
3. Window C: -349[ms] to 350[ms]
4. Window D: -999[ms] to 0[ms]
5. Window E: -499[ms] to 0[ms]
6. Window F: -349[ms] to 0[ms]
7. Window G: -249[ms] to 0[ms]
8. Window H: -99[ms] to 0[ms]

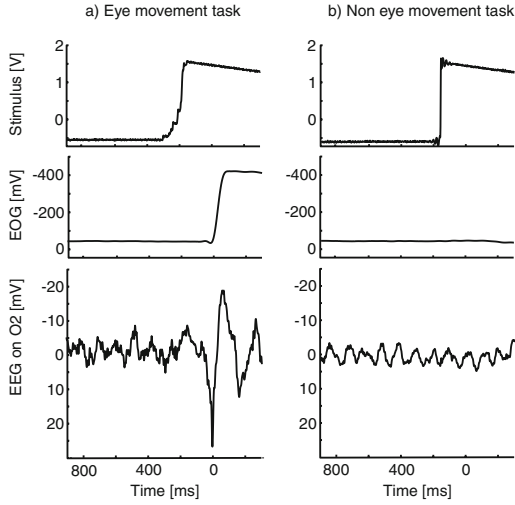


Fig. 2. Saccade-related EEG signals

0[ms] indicates the starting point of saccade. In order to observe influence of noises caused by EOG signals on EEG signals, we separated window size into two groups: Window A to C including EEG signals after saccade and window D to H excluding EEG signals after saccade.

In using five algorithms and eight windows, we calculated saccade-related independent components.

4 Saccade-Related EEG Signals

Fig. 2 indicates saccade-related EEG signals when a subject moves his/her eyes to the right side target. When a subject moves his/her eyes to the right side target, the EEG signal recorded on O2 (a electrode on right occipital lobe) changes sharply just before eye movements.

In this paper, we would like to extract this sharply changed EEG signal before eye movements.

5 Experimental Results

First, we define two words: an extracting rate and saccade-related IC. The extraction rate is defined by the following ratio:

$$\frac{\text{(the number of trials in which saccade-related IC are extracted)}}{\text{(The total number of trials)}}$$

We make assumption that a saccade-related IC has a positive peak from -50 [ms] \sim -1 [ms]. The peak-amplitude n is larger than 3; $n = \frac{\bar{x} - \mu}{s}$; where \bar{x} is mean of

EEG potential during 1000 [ms] before saccade, μ is maximum amplitude, and s is standard deviation during 1000 [ms] before saccade.

Table 1 represents the rate for extracting saccade-related ICs from the raw EEG data by each algorithm in the case of window E. From these results, the FastICA and JADE got good performances in extracting saccade-related independent components. However, the results of the AMUSE and SOBI and MILCA algorithm were not good. From these results, in order to extract saccade-related EEG signals, it is not suitable to use independency of 2nd order statistics and the mutual information.

Next, we focus on extracting rate in each windows (see Table 2). From Table 2 extracting rates in category A were lower than those in category B. Therefore, we should not use EEG signals after saccade. The signals in category A include EOG noise. This is reason for low extracting rate in category A. In the case of category B, the results of small window size is better. From these result, we can get good results in the case of short window size excluding signals after saccade.

6 Problem of Normal ICA Algorithms

In previous results, normal ICA algorithms can not extract saccade-related EEG signals in the case of using the window group A. In order to solve this problem, we must use EEG signals in category B. However, in real processing, we must calculate EEG signals influenced by EOG signals.

In order to calculate EEG signals influenced by EOG signals, we apply the EEG signals to a modified ICA algorithm. The modified ICA algorithm is base on Fast ICA because results of Fast ICA are better than results of another ICA algorithm. Input signals of the modified ICA algorithm are EEG signals and a reference signal. The modified ICA algorithm can extract an EEG signal related to a reference signal. For this approach, modified ICA algorithm can extract saccade-related EEG signals in the case of using the window group A.

7 Fast ICA with Reference Signal

The Fast ICA, one of the ICA algorithms, is based on a cost function minimization or maximization that is a function of the kurtosis $\kappa(\mathbf{w}^T \mathbf{x}) = \mathbf{E}(\mathbf{w}^T \mathbf{x})^4 - 3[\mathbf{E}\{\mathbf{w}^T \mathbf{x}\}^2]^2 = \mathbf{E}\{(\mathbf{w}^T \mathbf{x})^4\} - 3\|\mathbf{w}\|^4$; \mathbf{w} is one of the row of \mathbf{W}). Then Fast ICA changes the weight \mathbf{w} to extract an IC with the fixed-point algorithm.

Table 1. Extracted rate by four ICA algorithms

	AMUSE	FICA	JADE	SOBI	MILCA
A	14%	98%	100%	70%	50%
B	18%	82%	94%	76%	46%
C	30%	94%	96%	80%	62%
D	30%	98%	98%	66%	50%
E	24%	94%	96%	70%	46%

Table 2. Extracted rate by six window size

category	Window size	FastICA	JADE
A	-999 ~ 1000 [ms]	37.2%	38%
	-499 ~ 500 [ms]	29.6%	27.2%
	-349 ~ 350 [ms]	22.4%	26.4%
B	-999 ~ 0 [ms]	90%	93.6%
	-499 ~ 0 [ms]	93.2%	96.4%
	-349 ~ 0 [ms]	99.4%	99.2%
	-249 ~ 0 [ms]	93.2%	93.6%
	-99 ~ 0 [ms]	99.4%	99.2%

From among the several ICA algorithms, we selected the "Modified Fast ICA with Reference signal (FICAR)" algorithm to use in this study [10]. This algorithm can extract only the desired component by initializing the algorithm with prior information on the signal of interest. The main advantage of our approach is that users can give instructions to extract a desired signal correctly.

Fig. 3 shows an overview of the procedures of the proposed algorithm. First, the principal component analysis (PCA) outputs are calculated from original recorded signals to speed up the convergence of the algorithm. Second, this algorithm initializes \mathbf{w}_k ($k = 0$; k is the iteration number.) using some priori information included in a signal, \mathbf{d} , correlated with \mathbf{s}_i , i.e. $\mathbf{E}[\mathbf{d}\mathbf{s}_i] \neq 0$. This algorithm estimates a weight vector \mathbf{w} . Therefore, we calculate the error between \mathbf{d} , which is a reference signal, and $\mathbf{u} = \mathbf{w}^T \mathbf{x}$; $\varepsilon = \mathbf{d} - \mathbf{u}$. The weights are updated by the minimization of the mean-squared error (MSE) given by $\mathbf{E}[\varepsilon^2]$. To calculate the MSE, the least mean square (LMS) is used in order to calculate the MSE. After some calculations, the optimum weight (also called the Wiener weight) to minimize the MSE was found to be $\mathbf{w}^* = \mathbf{E}[\mathbf{d}\mathbf{x}]$. This algorithm initialized $\mathbf{w}_0 = \mathbf{E}[\mathbf{d}\mathbf{x}]/\|\mathbf{E}[\mathbf{d}\mathbf{x}]\|$. Third, this algorithm calculates \mathbf{w}_{k+1} by $\mathbf{w}_{k+1} = \mathbf{E}[\mathbf{x}(\mathbf{w}_k^T \mathbf{x})^3] - 3\mathbf{w}$ to maximize kurtosis. Then this algorithm can extract an IC closest to a reference signal or strictly speaking IC which is correlated with the reference signal.

8 Extraction Rate by FICAR

In the FICAR, the shape of the reference signal is that of an impulse signal having one peak. This shape is caused for two reasons. First, the saccade-related EEG has a sharp change like an impulse. Second, the main components of an EEG signal are the neural responses, and the waveform of the neural responses resembles an impulse.

We will determine the number of the saccade-related ICs obtained by using the FICAR. Table 3 represents the rate for extracting saccade-related ICs from the raw EEG data. The extraction rate is defined at the same as normal ICA algorithms. In this case, we use window A to window F.

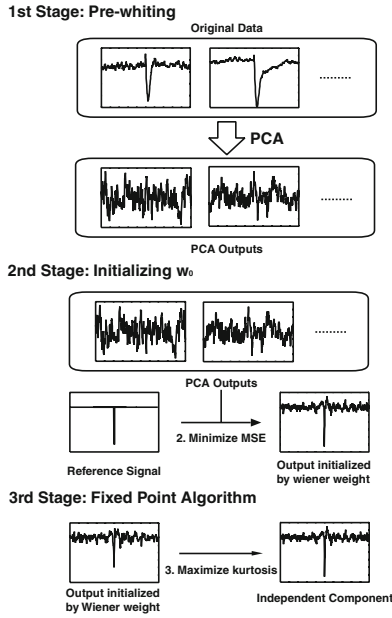


Fig. 3. Conceptual three stage for extraction desired ICs

Table 3. Extracted rate by six window size in case of FICAR

category	Window size	Subject					Ave.
		A	B	C	D	E	
A	-999 ~ 1000 [ms]	60%	60%	52%	80%	88%	68%
	-499 ~ 500 [ms]	68%	64%	60%	84%	88%	73%
	-349 ~ 350 [ms]	72%	68%	64%	88%	88%	76%
B	-999 ~ 0 [ms]	64%	68%	68%	84%	92%	75%
	-499 ~ 0 [ms]	72%	72%	72%	88%	96%	80%
	-349 ~ 0 [ms]	76%	80%	80%	92%	96%	85%

The lowest rate was 52%. However, the rates for most of the subjects were over 60% and the highest one was 96%. The average rate was 76.2%. In case of the window category B, extracting rates of the FICAR are better than extracting rates of the AMUSE, SOBI, and MILCA. However, the FastICA and JADE is better than extracting rate of the FICAR. In case of the window category A, extracting rate of the FastICA and JADE is from 22.4% and 38.0%(See Table 2). These results show that FICAR can extract saccade-related EEG signals much better than the FastICA and JADE, because the FICAR uses a reference signal and the FICAR can extract saccade-related ICs without influence of EOG noises.

9 Conclusion

This paper presented extraction rates of saccade-related EEG signals by five ICA algorithms and eight window sizes.

As results of extracting rate focused on ICA algorithms, The JADE and Fast ICA had good results.

As results of extracting rates focused on window sizes, the window H (-99[ms] ~ 0[ms]) had good results. In the case of the window A,B, and C, we could not get good results because these windows included big EOG noise.

In order to improve extracting rates in case of the window category A, we use improved FastICA with reference signal. In these results, we can confirm that extracting rates in case of the window category A are much higher than normal ICA algorithms.

In next step, we must check relationship between extracting rate and the number of input channels. In order to develop BCI, we must select a few input channels instead of present input channels; 19 channels.

References

1. Funase, A., Yagi, T., Kuno, Y., Uchikawa, Y.: A study on electro-encephalo-gram (EEG) in eye movement. *Studies in Applied Electromagnetics and Mechanics* 18, 709–712 (2000)
2. Cichocki, A., Amari, S.: *Adaptive blind signal and image processing*. Wiley, Chichester (2002)
3. Funase, A., Hashimoto, T., Yagi, T., Barros, A.K., Cichocki, A., Takumi, I.: Research for estimating direction of saccadic eye movements by single trial processing. In: *Proc. of 29th Annual International Conference of the IEEE EMBS*, pp. 4723–4726 (2007)
4. Hyvärinen, A., Oja, E.: A fast fixed-point algorithm for independent component analysis. *Neural Computation* (9), 1483–1492 (1997)
5. Choi, S., Cichocki, A., Amari, S.: Flexible independent component analysis. *Journal of VLSI Signal Processing* 26(1), 25–38 (2000)
6. Tong, L., Soon, V., et al.: Indeterminacy and indentifiability of blind indentification. *IEEE Trans. CAS* 38, 499–509 (1991)
7. Cardoso, J.-F., Souloumiac, A.: Blind beam-forming for non Gaussian signals. *IEE Proceedings-F* 140, 362–370 (1993)
8. Belouchrani, A., Abed-Meraim, K., Cardoso, J.F., Moulines, E.: Second-order blind separation of temporally correlated sources. In: *Proc. Int. Conf. on Digital Sig. Proc. (Cyprus)*, pp. 346–351 (1993)
9. Stogbauer, H., Kraskov, A., Astakhov, S.A., Grassberger, P.: Least Dependent Component Analysis Based on Mutual Information. *Phys. Rev. E* 70 (6), 066123 (2004)
10. Barros, A.K., Vigário, R., Jousmäki, V., Ohnishi, N.: Extraction of event-related signal form multi-channel bioelectrical measurements. *IEEE Transaction on Biomedical Engineering* 47(5), 61–65 (2001)

Learning of Mahalanobis Discriminant Functions by a Neural Network

Yoshifusa Ito¹, Hiroyuki Izumi², and Cidambi Srinivasan³

¹ School of Medicine, Aichi Medical University
Nagakute, Aichi-ken, 480-1195 Japan
`ito@aichi-med-u.ac.jp`

² Department of Policy Science, Aichi-Gakuin University
Nisshin, Aichi-ken, 470-0195 Japan
`hizumi@psis.aichi-gakuin.ac.jp`

³ Department of Statistics, University of Kentucky
Patterson Office Tower, Lexington, Kentucky 40506, USA
`srini@ms.uky.edu`

Abstract. It is known that a neural network can learn a Bayesian discriminant function. Ito et al. (2006) has pointed out that if the inner potential of the output unit of the network is shifted by a constant, the output becomes a Mahalanobis discriminant function. However, it was a heavy task for the network to calculate the constant. Here, we propose a new algorithm with which the network can estimate the constant easily. This method can be extended to higher dimensional classifications problems without much effort.

1 Introduction

The Mahalanobis discriminant function is commonly used for classification as alternative to the Bayesian discriminant function. The focus of this article relates to neural network with a single hidden layer which outputs the value of the Mahalanobis discriminant function.

Funahashi [2] proposed a single hidden layer neural network which can be trained to output the Bayesian discriminant for the two-category normal distribution case. We remarked in Ito et.al [7] that if the inner potential of the output unit of his network or its modification, which we have proposed [3-6], [8], [9], is shifted by a constant, then the resulting output can approximate the Mahalanobis discriminant function. The constant depends on the unknown covariance matrices and has to be estimated from the training data. However, the task of calculating the constant from the training data is difficult for the network and, in the simulation in [7], the constant was computed outside the network.

The goal of this paper is to propose a new algorithm which enables the neural network to estimate the constant easily. The network is equipped with an additional node for the estimation, and the training of the network is performed twice. The first training is to estimate the constant and the second to approximate the Bayesian discriminant function. The node stores the estimated constant and outputs it as steady bias of the inner potential of the output unit in the

second training. If, upon the completion of the second training, the connection from the node is removed then the inner potential is shifted by the constant and the output approximates the Mahalanobis discriminant function. This method, in principle, can be extended with ease to any higher dimensional case.

In our initial studies, the numerical optimization of the inner parameters associated with the activation functions of the hidden layer units turned out to be difficult in the higher dimensional case when the teacher signals are dichotomous. As a result, the simulations in our earlier articles [4-7] treated one-dimensional problems. However, we have been able to overcome this by refining the network to have fewer inner parameters, [3],[8],[9], and successfully extend the space of patterns to higher dimensions. Though this causes the number of hidden layer units to increase, the total number of parameters to be optimized is not considerably increased. In this paper the experimental results treat the two dimensional case.

2 Preliminaries

We treat the two-category normal-distribution case. The categories are denoted by θ_1 and θ_2 and we set $\Theta = \{\theta_1, \theta_2\}$. The patterns are from the d -dimensional Euclidean space \mathbf{R}^d . Denote by $N(\mu_i, \Sigma_i)$, $i = 1, 2$, the state-conditional probability distributions of the respective categories, where μ_i and Σ_i are the mean vectors and covariance matrices of the normal distributions. Their probability density functions are

$$p(x|\theta_i) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_i|}} e^{-\frac{1}{2}(x-\mu_i)^t \Sigma_i^{-1}(x-\mu_i)}, \quad i = 1, 2. \quad (1)$$

For simplicity, we suppose that the covariance matrices are not degenerate. Let $x, y \in \mathbf{R}^d$ be two patterns. The respective normal distributions define the Mahalanobis generalized distances between the two vectors by

$$d_i(x, y) = |(x - y)^t \Sigma_i^{-1}(x - y)|^{1/2}. \quad (2)$$

In the case of Mahalanobis discriminant analysis, if $d_1(x, \mu_1) < d_2(x, \mu_2)$, then the vector x is allocated to the category θ_1 and vice versa. Hence,

$$\psi_M(x) = -\frac{1}{2}\{d_1(x, \mu_1)^2 - d_2(x, \mu_2)^2\} \quad (3)$$

is a Mahalanobis discriminant function. If $\psi_M(x) > 0$, the vector x is allocated to the category θ_1 . By (2) and (3), we have

$$\psi_M(x) = -\frac{1}{2}\{(x - \mu_1)^t \Sigma_1^{-1}(x - \mu_1) - (x - \mu_2)^t \Sigma_2^{-1}(x - \mu_2)\}. \quad (4)$$

In the case of the Bayesian decision, the posterior probabilities are compared. Let $P(\theta_i)$ and $p(x|\theta_i)$, $i = 1, 2$, be the priors and the state-conditional probabilities of the respective categories. We set $p(x) = P(\theta_1)p(x|\theta_1) + P(\theta_2)p(x|\theta_2)$.

In the two-category case, one of the posterior probabilities, say $P(\theta_1|x)$, and the ratio $P(\theta_1|x)/P(\theta_2|x)$ are Bayesian discriminant functions. Since a monotone transform of a discriminant function is again a discriminant function [1],

$$\psi_B(x) = \log \frac{P(\theta_1|x)}{P(\theta_2|x)} = \log \frac{P(\theta_1)}{P(\theta_2)} + \log \frac{p(x|\theta_1)}{p(x|\theta_2)}. \tag{5}$$

is also a Bayesian discriminant function. If $\psi_B(x) > 0$, x is allocated to the category θ_1 . We remark that, though the discriminant functions (4) and (5) are based on distinct concepts, they differ only by a constant

$$C = \log \frac{P(\theta_1)}{P(\theta_2)} - \frac{1}{2} \log \frac{|\Sigma_1|}{|\Sigma_2|}. \tag{6}$$

Let σ be the logistic function: $\sigma(t) = (1 + e^{-t})^{-1}$. Since this is monotone,

$$\sigma(\psi_B(x)) = \sigma \left(\log \frac{P(\theta_1|x)}{P(\theta_2|x)} \right) = \frac{P(\theta_1|x)}{P(\theta_1|x) + P(\theta_2|x)} = P(\theta_1|x), \tag{7}$$

$$\sigma(\psi_M(x)) = \sigma \left(\log \frac{P(\theta_1|x)}{P(\theta_2|x)} - C \right) \tag{8}$$

are also a Bayesian discriminant function and a Mahalanobis discriminant function respectively. Note that the right-hand side of (7) is the posterior probability, implying it is the logistic transform of the quadratic function ψ_B as was remarked by Funahashi [3].

3 Training of the Neural Network

Let $F(x, w)$ be the output of a neural network with weight vector w . For an integrable function $\xi(x, \theta)$ defined on $\mathbf{R}^d \times \Theta$, let $E[\xi(x, \cdot)|x]$ and $V[\xi(x, \cdot)|x]$ be its conditional expectation and variance. The proposition below is proved in [11] and has been used by many authors [2], [4-10], [12].

Proposition. Set

$$\mathcal{E}(w) = \int_{\mathbf{R}^d} \sum_{i=1}^2 (F(x, w) - \xi(x, \theta_i))^2 P(\theta_i) p(x|\theta_i) dx. \tag{9}$$

Then,

$$\mathcal{E}(w) = \int_{\mathbf{R}^d} (F(x, w) - E[\xi(x, \cdot)|x])^2 p(x) dx + \int_{\mathbf{R}^d} V[\xi(x, \cdot)|x] p(x) dx. \tag{10}$$

If $\xi(x, \theta_1) = 1$ and $\xi(x, \theta_2) = 0$, then $E[\xi(x, \cdot)|x] = P(\theta_1|x)$. Hence, when $\mathcal{E}(w)$ is minimized, the output $F(x, w)$ is expected to approximate $P(\theta_1|x)$.

Let $G(x, w)$ be the inner potential of the output unit of the network. If $F(x, w)$ can approximate the posterior probability $P(\theta_1|x)$ with any accuracy

in $L^2(\mathbf{R}^d, p)$, then $G(x, w)$ can approximate the Bayesian discriminant function $\psi_B(x)$ in $L^2(\mathbf{R}^d, p)$ with any accuracy:

$$G(x, w) \doteq -\frac{1}{2}\{(x - \mu_1)^t \Sigma_1^{-1}(x - \mu_1) - (x - \mu_2)^t \Sigma_2^{-1}(x - \mu_2)\} + C, \tag{11}$$

where \doteq stands for approximation with any accuracy and C is defined by (6).

Accordingly, training of the network is carried out by minimizing

$$E_n(w) = \frac{1}{n} \sum_{t=1}^n (F(x^{(k)}, w) - \xi(x^{(k)}, \theta^{(k)}))^2, \tag{12}$$

where $\{(x^{(k)}, \theta^{(k)})\}_{k=1}^n \subset \mathbf{R}^d \times \Theta$ is the teacher sequence.

Since the teacher signals from the respective categories are paired with θ_i , the mean vectors μ_i of the patterns from the respective categories θ_i can be estimated by a simple gradient descent method. Set $y = x - \mu_i$ for x from the category θ_i , $i = 1, 2$. If the Bayesian neural network is trained with the sequence $\{(y, \theta_i)\}$, the inner potential of the output unit approximates

$$\psi_C(y) = -\frac{1}{2}\{y^t \Sigma_1^{-1}y - y^t \Sigma_2^{-1}y\} + C, \tag{13}$$

if learning goes well. Since

$$\psi_C(0) = C, \tag{14}$$

we have

$$G(0, w_0) \doteq C, \tag{15}$$

where w_0 is the weight vector of the network when the learning with $\{(y, \theta)\}$ is completed. Hence,

$$\psi_M(x) \doteq G(x, w) - G(0, w_0). \tag{16}$$

We first train the network with $\{(y, \theta_i)\}$. Then, the inner potential $G(0, w_0)$ is stored in the additional node. The second training is with $\{(x, \theta_i)\}$, during which the node is connected to the output unit and the memorized constant $G(0, w_0)$, an approximation of C , is fed into the output unit to bias the inner potential. When the second training is successfully completed, the output $F(x, w)$ of the network approximates the posterior probability $P(\theta_1|x)$ and the inner potential the Bayesian discriminant function $\psi_B(x)$. If the additional node is disconnected at this stage, the inner potential is shifted by $-G(0, w_0)$ and approximates the Mahalanobis discriminant function $\psi_M(x)$. Hence, the output approximates another Mahalanobis discriminant function $\sigma(\psi_M)$.

4 Simulations

Simulations are performed to confirm that the algorithm works well. By (1), (5) and (7), it is obvious that if the inner potential of the output unit can approximate any quadratic form, the network can approximate the Bayesian discriminant function $P(\theta_1|x)$. A one-hidden-layer neural network having $d + 1$ hidden

units has this capability, but we have experienced that optimization of the inner parameters of such a network is very difficult when the teacher signals are dichotomous random data. Hence, we are currently using networks having smaller numbers of inner parameters [3],[8],[9], for approximating Bayesian discriminant functions.

We use such network here as their approximation capability is guaranteed by our experience. The network is based on a theorem in [3]. The main theorem

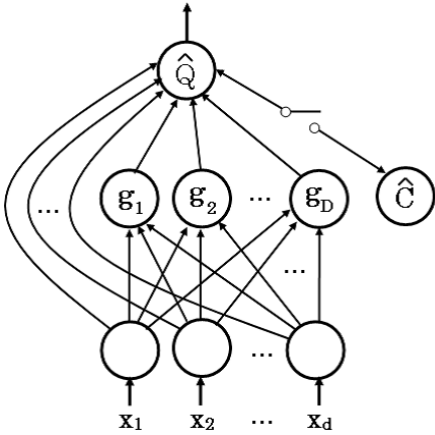


Fig. 1. A one-hidden-layer neural network having direct connections between the input layer and the output unit, and an additional node

states that any polynomial of degree n in \mathbf{R}^d can be approximated by

$$P(x) = \sum_{i=0}^n \sum_{k=1}^{D_i} a_{ik} g(\delta_i u_k \cdot x) \quad (17)$$

in the sense of $L^2(\mathbf{R}^d, p)$ if the probability measure p is rapidly decreasing. Here, $D_i = i+d-1 C_i$, u_k are unit vectors which can be fixed beforehand and g is an activation function which satisfies a mild condition. In (17) only δ_i are inner parameters to be optimized.

In this equation, one term is to approximate a constant and some others are to approximate a linear function. Hence, it is reduced to

$$Q(x) = \sum_{k=1}^D a_{2k} g(\delta_2 u_k \cdot x) + \sum_{k=1}^d v_k \cdot x + c \quad (18)$$

for $n = 2$, where $D = \frac{d(d+1)}{2}$. The network illustrated in Fig.1 can realize (18) without the annexed node \hat{C} . We use this network with the annexed node. The role of the node is to store the estimated value of the constant C as stated before. The direct connections between the input units and the output unit realize the linear sum $\sum v_k \cdot x$.

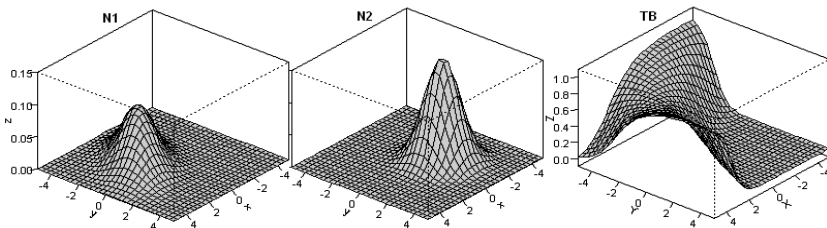


Fig. 2. N1, N2: State-conditional probability density functions of the respective categories. TB: The Bayesian discriminant function theoretically obtained.

In each simulation, 1000 patterns are randomly chosen from the two categories according to the prior probabilities. We present here the result of one of simulations we performed. Others will be presented elsewhere. In the simulation, the prior probabilities, mean vectors and covariance matrices are $P(\theta_1) = 0.4$, $P(\theta_2) = 0.6$, $\mu_1 = (1, -1)$, $\mu_2 = (0, 0)$, and $\Sigma_1 = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$, $\Sigma_2 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$.

The state-conditional probability densities of the respective categories and the theoretically obtained Bayesian discriminant function are illustrated in Figure 2. The teacher signals are pairs $\{(x^{(k)}, \theta^{(k)})\}_{k=1}^n$, $x^{(k)} \in \mathbf{R}^d$, $\theta^{(k)} \in \Theta$, $n = 1000$. They are generated independently according to the product probability measure on $\mathbf{R}^d \times \Theta$. Let $\{(x^{(k_{1i})}, \theta^{(k_{1i})})\}_{i=1}^{n_1}$ be a subsequence of the teacher signals, which includes all pairs from the category θ_1 . The sample mean vector of $x^{(k_{1i})}$ is $\hat{\mu}_1 = \frac{1}{n} \sum_{i=1}^{n_1} x^{(k_{1i})}$. We define $\{(x^{(k_{2i})}, \theta^{(k_{2i})})\}_{i=1}^{n_2}$ and $\hat{\mu}_2 = \frac{1}{n} \sum_{i=1}^{n_2} x^{(k_{2i})}$ in the same way. The centered teacher sequence is defined by replacing $x^{(k)}$ by $y^{(k)} = x^{(k)} - \hat{\mu}_i$, where $i = 1$ if $\theta^{(k)} = \theta_1$ and $i = 2$ if $\theta^{(k)} = \theta_2$. The network is trained first by $\{(y^{(k)}, \theta^{(k)})\}_{k=1}^n$ and then by $\{(x^{(k)}, \theta^{(k)})\}_{k=1}^n$.

When the first training is completed, its output is $F(y, w_0)$, where w_0 is defined in Section 3, and the inner potential of the output unit is $G(y, w_0)$ for the input y . The inner potential $G(0, w_0)$ for $y = 0$, an approximation of C , is memorized in the additional node and used as a bias of the inner potential during the second learning. When the second training is completed, the connection to the node is cut off. Then, the inner potential approximates $\psi_M(x)$ and the output approximates the Mahalanobis discriminant function $\sigma(\psi_M(x)) \approx \sigma(G(x, w) - G(0, w_0))$.

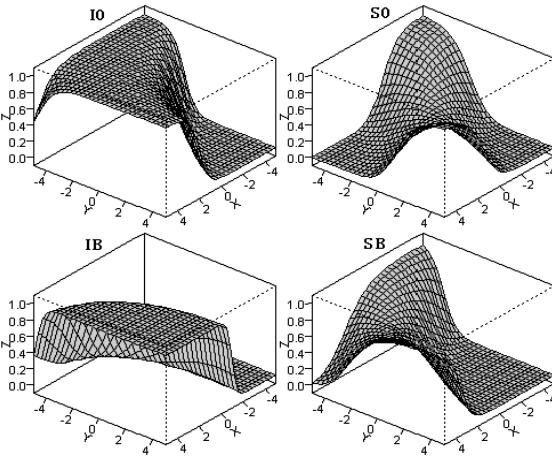


Fig. 3. The initial (**IO** and **SO**) and final (**SI** and **SB**) patterns of the outputs at the two trainings

Fig. 3 illustrates the outputs of the network: **IO** and **IB** are the initial patterns of the outputs at the first and second trainings, and **SO** and **SB** are the outputs when the respective learnings are completed. The initial patterns are rather arbitrarily chosen. Even if they are interchanged, the patterns they converge to are almost the same as **SO** and **SB**.

Though the real value of C is -0.2616, the estimated value is $G(0, w_0) = -0.2492$. The small error may be due to the approximation error of the network and the deviation of the empirical distribution from the given probability distributions. In the simulation, this estimated value is stored in the additional node

and used. Fig.4 illustrates the Mahalanobis discriminant functions $\sigma(\psi_M)$ (**TM**) theoretically obtained, the Mahalanobis discriminant function (**SM**) obtained by simulation, and their difference (**TM-SM**).

As a test sequence, we used 1000 pairs generated independently of the teacher sequence. The test sequences contained 408 patterns from the category θ_1 and 592 from the category θ_2 . The classification results by the four discriminant functions are shown in Table 1a. The numbers in the **TM**, **SM**, **TB** and **SB** columns in Table 1 are respectively the classification results by the Mahalanobis discriminant functions obtained theoretically (**TM**) and by simulation (**SM**), and those by the Bayesian discriminant functions obtained theoretically (**TB**) and by simulation (**SB**). The numbers in the first row (Alloc. to θ_1) are those of the patterns allocated to the category θ_1 , and the numbers in the second row (Correct. Alloc.) are those of the patterns correctly allocated. Among 1000 allocations by **SM** (**SB**), 987 (989) coincided with those by **TM** (**TB**) as listed in Table 1b. The allocation capabilities of the simulated discriminant functions are comparable to the corresponding theoretical discriminant functions respectively.

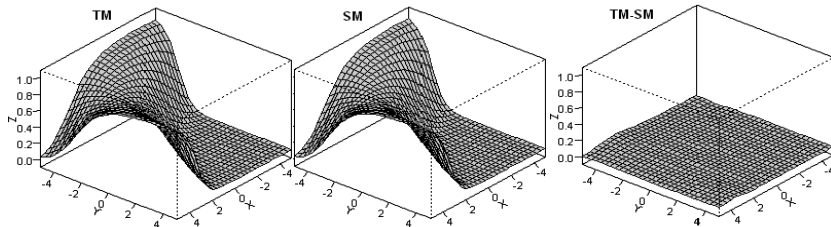


Fig. 4. TM, SM: Mahalanobis discriminant functions obtained theoretically and by simulation. **TM-SM:** difference between **TM** and **SM**.

Table 1. a: Allocation by neural networks. **b:** Numbers of identical allocations by theoretical discriminant functions and neural networks.

	TM	SM	TB	SB		TM = SM	TB = SB
Alloc. to θ_1	454	441	521	510	Ident. Alloc.	987	989
Correct Alloc.	704	699	707	704			

a

b

5 Discussions

In Ito et al. [7] we have remarked that the inner potential of the output unit of the trained Bayesian neural network can approximate a Mahalanobis discriminant function when shifted by the constant C . However, the calculation of C was a heavy burden for the neural network, in particular, in the case of higher dimensional pattern classification. The algorithm proposed in this paper alleviates this burden.

In [7], simulations were done only in the case of one-dimensional patterns due to the limited capability of the network. The limit came from the difficulty not only in calculating C but also in optimizing the inner parameters. This time, we used a neural network with a smaller number of inner parameters based on Ito [3]. As a result, the space of the patterns are extended to the two-dimensional space.

The network used here is a modification of Funahashi's for the two-category normal distribution case [2]. However, in applications, the data are not necessarily from normal populations. In such cases, the discriminant function obtained by sifting the Bayesian discriminant function can deviate from the Mahalanobis discriminant function. One way to avoid this may be to restrict the space of functions which the inner potential of the output unit can approximate. If the activation function of the hidden layer units is replaced by a parabolic function t^2 , the inner potential is restricted to quadratic forms.

References

1. Duda, R.O., Hart, P.E.: Pattern classification and scene analysis. John Wiley & Sons, New York (1973)
2. Funahashi, K.: Multilayer neural networks and Bayes decision theory. *Neural Networks* 11, 209–213 (1998)
3. Ito, Y.: Simultaneous approximations of polynomials and derivatives and their applications to neural networks. *Neural Computation* 20, 2757–2791 (2008)
4. Ito, Y., Srinivasan, C.: Multicategory Bayesian decision using a three-layer neural network. In: Kaynak, O., Alpaydm, E., Oja, E., Xu, L. (eds.) ICANN 2003 and ICONIP 2003. LNCS, vol. 2714, pp. 253–261. Springer, Heidelberg (2003)
5. Ito, Y., Srinivasan, C.: Bayesian decision theory on three-layer neural networks. *Neurocomputing* 63, 209–228 (2005)
6. Ito, Y., Srinivasan, C., Izumi, H.: Bayesian learning of neural networks adapted to changes of prior probabilities. In: Duch, W., Kacprzyk, J., Oja, E., Zadrozny, S. (eds.) ICANN 2005. LNCS, vol. 3697, pp. 253–259. Springer, Heidelberg (2005)
7. Ito, Y., Srinivasan, C., Izumi, H.: Discriminant analysis by a neural network with Mahalanobis distance. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4132, pp. 350–360. Springer, Heidelberg (2006)
8. Ito, Y., Srinivasan, C., Izumi, H.: Learning of Bayesian discriminant functions by a neural network. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) ICONIP 2007, Part I. LNCS, vol. 4984, pp. 238–247. Springer, Heidelberg (2007)
9. Ito, Y., Srinivasan, C., Izumi, H.: Multi-category Bayesian decision by neural networks. In: Kůrková, V., Neruda, R., Koutník, J. (eds.) ICANN 2008, Part I. LNCS, vol. 5163, pp. 21–30. Springer, Heidelberg (2008)
10. Richard, M.D., Lipmann, R.P.: Neural network classifiers estimate Bayesian a posteriori probabilities. *Neural Computation* 3, 461–483 (1991)
11. Ruck, M.D., Rogers, S., Kabrisky, M., Oxley, H., Sutter, B.: The multilayer perceptron as approximator to a Bayes optimal discriminant function. *IEEE Transactions on Neural Networks* 1, 296–298 (1990)
12. White, H.: Learning in artificial neural networks: A statistical perspective. *Neural Computation* 1, 425–464 (1989)

Implementing Learning on the SpiNNaker Universal Neural Chip Multiprocessor

Xin Jin, Alexander Rast, Francesco Galluppi,
Mukaram Khan, and Steve Furber

School of Computer Science, University of Manchester
Manchester, UK M13 9PL
{jinxa,rasta,francesco.galluppi,khanm}@cs.man.ac.uk,
steve.furber@manchester.ac.uk
<http://www.cs.manchester.ac.uk/apt>

Abstract. Large-scale neural simulation requires high-performance hardware with on-chip learning. Using SpiNNaker, a universal neural network chip multiprocessor, we demonstrate an STDP implementation as an example of programmable on-chip learning for dedicated neural hardware. Using a scheme driven entirely by pre-synaptic spike events, we optimize both the data representation and processing for efficiency of implementation. The deferred-event model provides a reconfigurable timing record length to meet different accuracy requirements. Results demonstrate successful STDP within a multi-chip simulation containing 60 neurons and 240 synapses. This optimisable learning model illustrates the scalable general-purpose techniques essential for developing functional learning rules on general-purpose, parallel neural hardware.

Keywords: Neural, Spiking, SpiNNaker, Learning, Event-Driven, STDP.

1 Introduction

Neural networks are intrinsically learning systems. Therefore hardware designed to support neural networks ought also to support learning. Nonetheless, many neural network chips have opted not to support any on-chip learning, because of scalability concerns revolving around complex update circuitry [1]. An even more fundamental limitation of most fixed neural hardware model is that it can support only one or at most a few selected families of neural network. A universal neural network device is necessary to develop large networks without prior commitment to a particular model. Such a device must have general-purpose support for on-chip learning. Furthermore as a result of not being “hard-wired” it can mitigate scalability concerns, exchanging expensive update circuits for simpler general-purpose synaptic logic. For such an architecture, there are three principal requirements: 1) that the device have specific dedicated programmable hardware that the model can use to implement learning, 2) that the learning rule itself be purely software or configuration commands, 3) that the learning implementation be efficient enough to realize the gains of hardware in a scalable

way. In the SpiNNaker chip, which is an example of a universal neural network chip, the previously-introduced virtual synaptic channel circuitry [2] provides generalized support for on-chip learning without constraining the learning rule. Using deferred-event processing to reorder events makes possible an efficient software-based, event-driven implementation of the well-known STDP learning rule. The methodology has the further advantage of being efficient both in memory utilization and processing overhead, since it only requires update on receipt of a presynaptic spike. The methods we show here translate theoretical learning rules into efficient implementations, and provide a path for future development of (possibly as yet undiscovered) learning rules in hardware.

2 Architecture and Models

2.1 The STDP Model

The model we consider is the Gerstner spike-timing-dependent-plasticity (STDP) learning rule [3], a Hebbian model for spiking neural networks. We use a simplified implementation (equation 1 and Figure 1(a)) [4].

$$F(\Delta t) = \begin{cases} A_+ e^{\frac{\Delta t}{\tau_+}} & \Delta t < 0, \\ -A_- e^{\frac{-\Delta t}{\tau_-}} & \Delta t \geq 0. \end{cases} \quad (1)$$

$$\Delta W = \varepsilon \sum_{pre} [\gamma + \sum_{post} F(\Delta t)] \quad (2)$$

Where Δt is the time difference between the pre- and post-synaptic spike timing, A_+ and A_- are the maximum amount of synaptic modification, τ_+ and τ_- are the time windows determining the range of spike interval over which the STDP

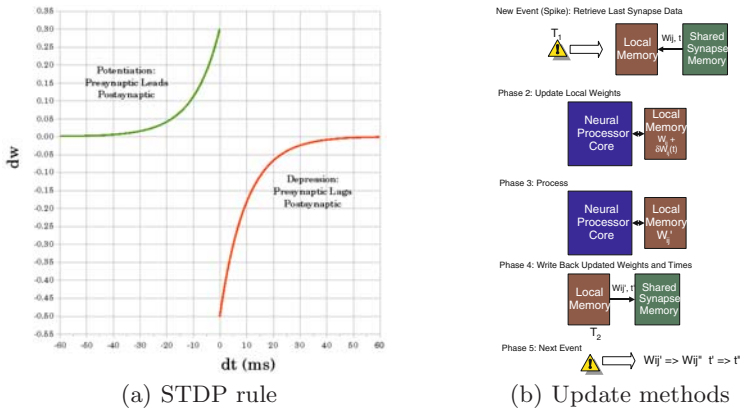


Fig. 1. STDP update rule: theoretical update curves and conceptual implementation

occurs. If the pre-synaptic spike arrives before the post-synaptic neuron fires (i.e. $t > 0$), it causes long-term potentiation (LTP) and the synaptic weight is strengthened according to A_+e^{-t/τ_+} . If the pre-synaptic spike arrives after the post-synaptic neuron fires (i.e. $t < 0$), it causes long-term depression (LTD) and the synaptic weight is weakened according to A_-e^{t/τ_-} . The modification is accumulated and the weight is updated according to equation 2.

2.2 SpiNNaker and the Event-Driven Model

As previously introduced [5], SpiNNaker is a universal neural network chip for massively-parallel real-time large-scale simulation. Without attempting to describe all the features that have been the subject of previous publication, three important design aspects are critical to the on-chip learning model. First, mapping of neurons to processors is many-to-one. Each ARM968 processor is capable of modeling upto 1000 Izhikevich neurons [6] with 1 millisecond time resolution in 16-bit fixed-point arithmetic [7]. Second, local memory resources are limited: a 64KB private data Tightly-Coupled Memory (TCM) is available to each processor; but global memory resources are large: a 1Gb external shared SDRAM is available to all 20 processors on a given chip. A dedicated DMA controller makes global memory “virtually local” to each processor by swapping data between SDRAM and TCM [2]. Most synaptic data therefore usually resides off-chip (and off-processor), the synaptic channel providing “just-in-time” local access.

Third, and most importantly, SpiNNaker uses an event-driven processing model with annotated real-time model delays [8]. There are two important events from the point of view of the model. A Timer event, occurring nominally each millisecond, drives neural state update. A spike event, occurring (asynchronously) whenever an input Address-Event-Representation (AER) spike packet arrives at a neuron, triggers synaptic state update. This event model makes it possible, by exploiting the difference between model “real” time and electronic “system” time, to reorder processing and redistribute synaptic memory resources in order to achieve efficient, yet accurate, on-chip learning [8].

The earlier work [8] outlines the basic method of the deferred event model. Key details of the implementation optimise learning for the hardware. Neurons are mapped to cluster groups (fascicles) of postsynaptic target neurons connecting to the same pre-synaptic source onto a single processor. Not only does this improve routability, but it allows a single contiguous memory area (called the synapse block) to contain all the synaptic information for the group. A synapse block is a compressed line of 32-bit words containing a 4-bit synaptic delay, a 12-bit postsynaptic index and a 16-bit synaptic weight. A single event therefore retrieves the entire group using a DMA operation and makes the entire block of synapses locally available to its respective processor. This permits the characteristic feature and most significant optimization of the method: *synaptic update occurs only upon presynaptic input events.*

3 Methodology

3.1 Mapping STDP to SpiNNaker

Most STDP implementations trigger weight update both on pre- and post-synaptic spikes [9, 10]. In this approach, calculating the Δt is simply a matter of comparing the history records of spiking timings. This corresponds to examining the *past* spike history (as in Figure 2(a)), at least within the STDP sensitivity window. However, in SpiNNaker, since the synapse block is a neuron-associative memory array, it can only be indexed either by the pre- or the post-synaptic neuron. If synapses are stored in pre-synaptic order, LTD will be very efficient while LTP plasticity will be inefficient, and vice versa - because one or the other lookup would require a scattered traverse of discontinuous areas of the synaptic block. Furthermore, because of the virtual synaptic channel memory model, a given pre-synaptic indexed synapse block will only appear in the TCM when an associated pre-synaptic spike arrives. As a result, a pre-post sensitive scheme would double the number of SDRAM accesses and be only partially able to take advantage of block-orientated contiguous burst transfers.

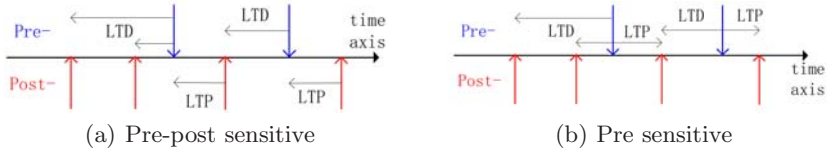


Fig. 2. STDP implementation methods

To solve this problem, we develop an alternative scheme: pre-synaptic sensitive update. The pre-synaptic sensitive scheme only triggers STDP with the arrival of pre-synaptic spikes (Figure 2(b)). This guarantees that the synapse block is always in the TCM when STDP is triggered, and makes accessing individual synapses possible by efficient iteration through the array elements when the synapse block is in pre-synaptic order. However, this requires examining not only the past spike history records, but also the future records. Naturally, future spike timing information is not available at the time the pre-synaptic spike arrives since it has not yet happened. The deferred-event model solves this problem by reordering the spike timing and performing STDP in the future (the current time plus the maximum delay and the time window). This ensures accurate recording and incorporation of future spike timings in the update.

3.2 Synaptic Delay and Timing Records

Synaptic delays (axonal conduction delays) play an important role in the simulation of spiking neural networks with plasticity. In SpiNNaker, delays are annotated as a post-process upon receipt of a spike, the individual delay values

being a synaptic parameter. This makes the delay itself entirely programmable; the reference model uses delays from 1 - 16 ms for each connection [7]. STDP requires both pre-synaptic and post-synaptic spike timings. The SDRAM stores a pre-synaptic time stamp with 2ms resolution at the beginning of each synapse block (Figure 3) which is updated when an associated spike arrives. The time stamp has two parts, a coarse time and fine time. Coarse time is a 32-bit digital value representing the last time the neuron fired. Fine time is a bitmapped field of 24 bits representing spike history in the last 48 ms. Post-synaptic time stamps reside in local TCM (Figure 3) and have a similar format to pre-synaptic time stamps except that they are 64 bits long (representing 128ms), allowing longer history records to account for input delays. Post-synaptic time stamps are updated when their corresponding neurons fire.

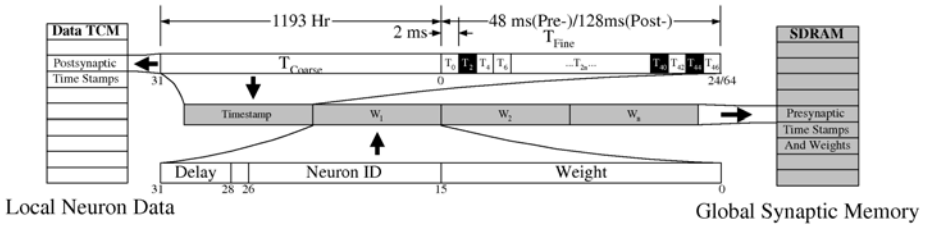


Fig. 3. Time stamps for STDP

3.3 Method and Model

Input pre-synaptic spikes trigger the learning rule following an algorithm that proceeds in three steps: update the pre-synaptic time stamp, traverse post-synaptic connections and update synaptic weights, as shown in Figure 1(b).

Step 1: Update the pre-synaptic time stamp. Firstly, the pre-synaptic time stamp is updated. The fine time stamp is shifted left until bit 0 equals the time of the current spike. If any ‘1’ is shifted out (going to bit 25), STDP starts. Bit 25 then represents the pre-synaptic spike time used to compute the update.

Step 2: Traverse post-synaptic connections. This step checks the post-synaptic connections one by one. First, the time of bit 25 is incremented by the synaptic delay to convert the electronic timing to the neural timing T . Second, the neuron’s ID is used as an index to retrieve the post-synaptic spike time stamp from the TCM.

Step 3: Update synaptic weights. Next, the processor calculates the LTD window $[T - T_-, T]$ and the LTP window $[T, T + T_+]$. If any bit in the post-synaptic time stamp is ‘1’ within the LTD window or LTP window, the synaptic weight is either potentiated or depressed according to the STDP rule.

Each of the three steps may run through several iterations. If there are n ‘1’s shifted to bit 25 in step 1, m connections in the synapse block in step 2 and l

bits within the time window in step 3, the computational complexity in Step 3 will dominate as $O(nml)$. For the sake of performance, Step 3 updates should be as efficient as possible.

3.4 Length of Time Stamps

The length of the time stamp effects both the performance and the precision of the STDP rule. Longer history records permit better precision at the cost of significantly increased computation time. Determining the optimal history length is therefore dependent upon the required precision and performance. The test model assumes peak firing rates of $\sim 10\text{Hz}$. TCM memory limitations lead to the choice of a 64-bit post-synaptic time stamp, able to record a maximum of 128ms. A 24-bit pre-synaptic time stamp with 2 ms resolution and a maximum of 16 ms delay guarantees a $24 * 2 - 16 = 32\text{ms}$ LTP window for any delay. This in turn permits a $1000/(128 - 32) = 10.4\text{Hz}$ firing rate to guarantee the same 32ms time window for LTD. These lengths are reconfigurable (dynamically if necessary) to any other value to meet different requirements.

4 Results

We implemented a neural network on a cycle accurate four-chip SpiNNaker simulator based on the ARM SOC designer [11] to test our model. The network is largely based on the code published in [10], which was also used to test the

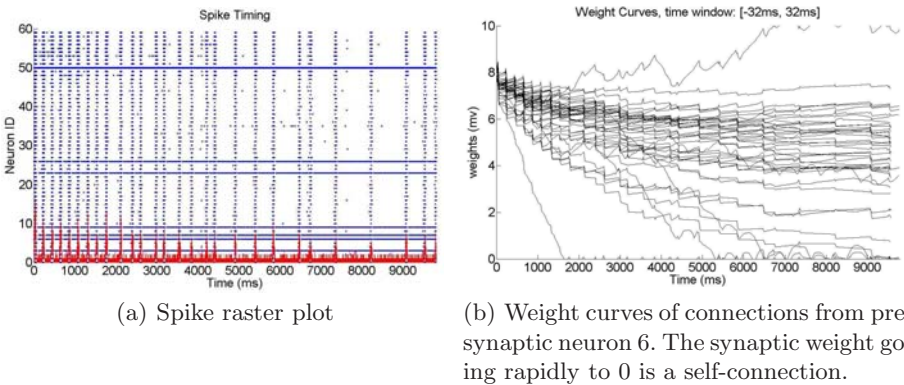


Fig. 4. STDP results. At the beginning of the simulation input neurons fire synchronously, exciting the network which exhibits high-amplitude synchronized rhythmic activity around 5 to 6 Hz. As synaptic connections evolve according to STDP, uncorrelated synapses are depressed while correlated synapses are potentiated. Since the network is small and the firing rate is low, most synapses will be depressed (as per panel b), leading to a lower firing rate. The synaptic weight going rapidly to zero is the self-connection of neuron 6: since each pre-synaptic spike arrives shortly after the post-synaptic spike the synapse is quickly depressed.

consistency of our results. It has 48 Regular Spiking Excitatory neurons ($a = 0.02$, $b = 0.2$, $c = -65$, $d = 8$) and 12 Fast Spiking Inhibitory neurons ($a = 0.1$, $b = 0.2$, $c = -65$, $d = 2$). Each neuron connects randomly to 40 neurons (self-synapses are possible) with random 1-16 ms delays; inhibitory neurons only connect to excitatory neurons. Initial weights are 8 and -4 for excitatory and inhibitory connections respectively. We used $\tau_+ = \tau_- = 32ms$, $A_+ = A_- = 0.1$ for STDP. Inhibitory connections are not plastic [12]. There are 6 excitatory and 1 inhibitory input neurons, receiving constant input current $I = 20$ to maintain a high firing rate. We ran the simulation for 10 sec (biological time). Figure 4 gives

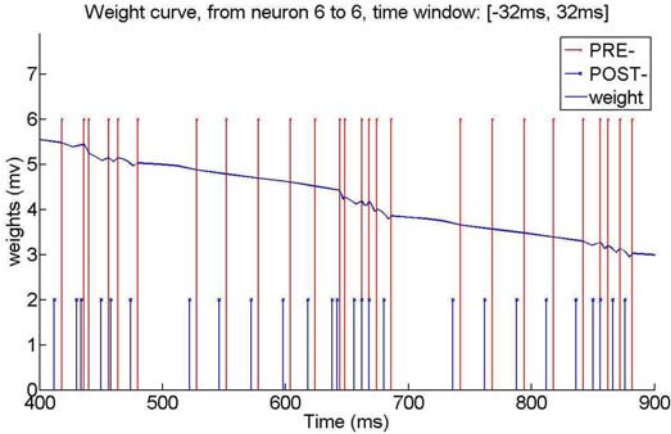


Fig. 5. Weight modification caused by the correlation of the pre and post time. Modification is triggered by pre-synaptic spikes. The weight curve in between two pre-synaptic spikes is firstly depressed because of LTD window and then potentiated because of the LTP window.

the results: the left part shows the raster plot and the right part the evolution of synaptic weights of connections from pre-synaptic neuron id 6 (an input neuron). Detailed modifications of the self-connection weight is shown in Figure 5 along with pre- and post-synaptic timing.

5 Discussion and Conclusion

Implementing STDP on SpiNNaker indicates that general-purpose neural hardware with on-chip, real-time learning support is feasible. The pre-synaptic sensitive scheme and the deferred-event model provide the core of the solution, but nonetheless as we have seen it requires careful optimization and an efficient implementation if it is to be effective. Implementing learning on any hardware neural system is a trade-off between performance and functionality. With SpiNNaker, the user can choose that trade-off according to the needs of their model.

There is considerable work remaining to develop both additional rules and additional extensions to the rule above. Besides maximizing performance and

accuracy with parameter adjustments, we are also investigating methods to implement chemical-dependent LTP and LTD (as well as methods for long-distance chemical transmission). The long-term goal is to have a “library” of learning rules that the user can instantiate on-chip or use as templates to modify in order to fit their model.

Acknowledgements. We would like to thank the Engineering and Physical Sciences Research Council (EPSRC), Silistix, and ARM for support of this research. S.B. Furber is the recipient of a Royal Society Wolfson Merit Award.

References

1. Maguire, L., McGinnity, T.M., Glackin, B., Ghani, A., Belatreche, A., Harkin, J.: Challenges for large-scale implementations of spiking neural networks on fpgas. *Neurocomputing* 71(1-3), 13–29 (2007)
2. Rast, A., Yang, S., Khan, M.M., Furber, S.B.: Virtual Synaptic Interconnect Using an Asynchronous Network-on-Chip. In: *Proc. 2008 Int’l Joint Conf. Neural Networks (IJCNN2008)*, pp. 2727–2734 (2008)
3. Gerstner, W., Kempter, R., van Hemmen, J.L., Wagner, H.: A Neuronal Learning Rule for Sub-millisecond Temporal Coding. *Nature* 383(6595), 76–78 (1996)
4. Song, S., Miller, K.D., Abbott, L.F.: Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience* 3, 919–926 (2000)
5. Plana, L.A., Furber, S.B., Temple, S., Khan, M.M., Shi, Y., Wu, J., Yang, S.: A GALS Infrastructure for a Massively Parallel Multiprocessor. *IEEE Design & Test of Computers* 24(5), 454–463 (2007)
6. Izhikevich, E.: Simple Model of Spiking Neurons. *IEEE Trans. Neural Networks* 14, 1569–1572 (2003)
7. Jin, X., Furber, S.B., Woods, J.V.: Efficient Modelling of Spiking Neural Networks on a Scalable Chip Multiprocessor. In: *Proc. 2008 Int’l Joint Conf. Neural Networks (IJCNN 2008)*, pp. 2812–2819 (2008)
8. Rast, A., Jin, X., Khan, M.M., Furber, S.: The Deferred Event Model for Hardware-Oriented Spiking Neural Networks. In: Köppen, M., Kasabov, N., Coghil, G. (eds.) *ICONIP 2008*. LNCS, vol. 5507, pp. 1057–1064. Springer, Heidelberg (2009)
9. Masquelier, T., Guyonneau, R., Thorpe, S.J.: Competitive STDP-based spike pattern learning. *Neural Computation* 21(5), 1259–1276 (2009)
10. Izhikevich, E.: Polychronization: Computation with spikes. *Neural Computation* 18(2), 245–282 (2006)
11. Khan, M., Painkras, E., Jin, X., Plana, L., Woods, J., Furber, S.: System level modelling for SpiNNaker CMP system. In: *Proc. 1st International Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools RAPIDO 2009* (2009)
12. Bi, G., Poo, M.: Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neuroscience* 18(24), 10464–10472 (1998)

Learning Gaussian Process Models from Uncertain Data

Patrick Dallaire, Camille Besse, and Brahim Chaib-draa

DAMAS Laboratory,
Computer Science and Software Engineering Department,
Laval University, Canada
{dallaire, besse, chaib}@damas.ift.ulaval.ca
<http://www.damas.ift.ulaval.ca>

Abstract. It is generally assumed in the traditional formulation of supervised learning that only the outputs data are uncertain. However, this assumption might be too strong for some learning tasks. This paper investigates the use of Gaussian Process prior to infer consistent models given uncertain data. By assuming a Gaussian distribution with known variances over the inputs and a Gaussian covariance function, it is possible to marginalize out the inputs' uncertainty and keep an analytical posterior distribution over functions. We demonstrated the properties of the method on a synthetic problem and on a more realistic one, which consist in learning the dynamics of the well-known cart-pole problem and compare the performance versus a classic Gaussian Process. A large improvement of the mean squared error is presented as well as the consistency of the result of the regression.

Keywords: Gaussian Processes, Noisy Inputs, Dynamical Systems.

1 Introduction

As soon as a regression has to be done using a statistical model on noisy inputs, the resulting quality of the estimated model may suffer if no attention is paid to the uncertainty of the training set. Actually, this may occur in two different ways, one due to the training with noisy inputs and the other due to an extra noise in the outputs caused by the noise over the inputs.

Statisticians already have investigated this problem in several ways: “total least-squares” [1] changes the cost of the regression problem to encourage the regressor to minimize both error due to noise on outputs as well as noise on inputs; the “error-in-variables” model [2] deals directly with noisy inputs by creating correlated virtual variables that thus have correlated noises. Recent work in machine learning has also addressed this problem, either by attempting to learn the entire input distribution [3], by integrating over chosen noisy points using estimated distribution during training [4] or by de-noising the inputs by accounting for the noise while training the model [5].

In this paper, we investigate an approach, pioneered by Girard [6], in which more than trying to *predict using noisy inputs*, we *learn from these inputs* by marginalizing out the inputs' uncertainty and keep an analytical posterior distribution over functions. This approach achieves two goals: First it shows that we are able to learn and make prediction from noisy inputs. Second, this method is applied to a well-known problem

of balancing a pole over a cart where the problem is to learn the 5-dimensional nonlinear dynamics of the system. Results show that taking into account the uncertainty of the inputs make the regression consistent and reduce drastically the mean squared error.

This paper is structured as follows. First, we formalize the problem of learning with noisy inputs and introduce some notations about Gaussian Processes and the regression model. In section 3 we present the experimental results on a difficult artificial problem and on a more realistic problem. Section 4 discusses the results and concludes the paper.

2 Preliminaries

A Gaussian Process (GP) is a stochastic process which is used in machine learning to describe a distribution directly into the function space. It also provides a probabilistic approach to the learning task and has the interesting property to give uncertainty estimates while doing predictions. The interested reader is invited to refer to [7] for more information on GPs.

2.1 Gaussian Process Regression

By using a GP prior, it is assumed that the joint distribution of the finite set of observations given their inputs is multivariate Gaussian. Thus, a GP is fully specified by its mean and covariance functions. Assume that a set of training data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ is available where $\mathbf{x}_i \in \mathbb{R}^D$, y is a scalar observation such that

$$y_i = f(\mathbf{x}_i) + \epsilon_i \quad (1)$$

and where ϵ_i is a white Gaussian noise. For convenience, we will use the notation $X = [\mathbf{x}_1, \dots, \mathbf{x}_N]$ for inputs and $\mathbf{y} = [y_1, \dots, y_N]$ for outputs. Under the GP prior model with zero mean function, the joint distribution of the training set is $\mathbf{y}|X \sim \mathcal{N}(\mathbf{0}, K)$ where K is the covariance matrix whose entries K_{ij} are given by the covariance function $C(\mathbf{x}_i, \mathbf{x}_j)$. This multivariate Gaussian probability distribution over the training observations can be used to compute the posterior distribution over functions. Therefore, making prediction is done by using the posterior mean and its associated measure of uncertainty, given by the posterior covariance. For a test input \mathbf{x}_* , the posterior distribution is $f_*|\mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}(\mu(\mathbf{x}_*), \sigma^2(\mathbf{x}_*))$ with mean and variance functions given by

$$\mu(\mathbf{x}_*) = \mathbf{k}_*^\top K^{-1} \mathbf{y} \quad (2)$$

$$\sigma^2(\mathbf{x}_*) = C(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top K^{-1} \mathbf{k}_* \quad (3)$$

where \mathbf{k}_* is the $N \times 1$ vector of covariance between \mathbf{x}_* and training inputs X . Although many covariance functions can be used to define a GP prior, we will use for the remainder of this paper the squared exponential which is one of the most widely used kernel function. The chosen kernel function

$$C(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp(-(\mathbf{x}_i - \mathbf{x}_j)^\top W^{-1} (\mathbf{x}_i - \mathbf{x}_j)) + \sigma_\epsilon^2 \delta_{ij} \quad (4)$$

is parameterized by a vector of hyperparameters $\theta = [W, \sigma_f^2, \sigma_\epsilon^2]$, where W is the diagonal matrix of characteristic length-scale, which account for different covariance

measure for each input dimension, σ_f^2 is the signal variance and σ_ϵ^2 is the noise variance. Varying these hyperparameters influence the interpretation of the training data by modifying the shapes of functions allowed by the GP prior. It might be difficult a priori to fix the hyperparameters of a kernel function and expect these to fit the observed data correctly. A common way to estimate the hyperparameters is to maximize the log likelihood of the observations \mathbf{y} [7]. The function to maximize is

$$\log p(\mathbf{y}|X, \theta) = -\frac{1}{2}\mathbf{y}^\top K^{-1}\mathbf{y} - \frac{1}{2}\log |K| - \frac{N}{2}\log 2\pi \tag{5}$$

since the joint distribution of the observations is a multivariate Gaussian. The maximization can be done using conjugate gradient methods to find an acceptable local maxima.

2.2 Learning with Uncertain Inputs

As we suit in the introduction, the assumption that only the outputs are noisy is not enough for some learning task. Consider the case where the inputs are uncertain and where each input value comes with variance estimates. It has been shown by Girard [6] that, for normally distributed inputs and using the squared exponential as kernel function, integrate over the input distribution analytically is feasible.

Consider the case where inputs are a set of Gaussian distributions rather than a set of point estimates. Therefore, the true input value \mathbf{x}_i is not observable, but we have access to its distribution $\mathcal{N}(\mathbf{u}_i, \Sigma_i)$. Thus, accounting for these inputs distributions is done by solving

$$C_n = \int \int C(\mathbf{x}_i, \mathbf{x}_j)p(\mathbf{x}_i)p(\mathbf{x}_j)d\mathbf{x}_i d\mathbf{x}_j \tag{6}$$

where $p(\mathbf{x}_i) = \mathcal{N}(\mathbf{u}_i, \Sigma_i)$ and $p(\mathbf{x}_j) = \mathcal{N}(\mathbf{u}_j, \Sigma_j)$. Since [8] involve integrations over products of Gaussians, the resulting kernel function is computed exactly with

$$C_n((\mathbf{u}_i, \Sigma_i), (\mathbf{u}_j, \Sigma_j)) = \frac{\sigma_f^2 \exp((\mathbf{u}_i - \mathbf{u}_j)^\top (W + \Sigma_i + \Sigma_j)^{-1} (\mathbf{u}_i - \mathbf{u}_j))}{|I + W^{-1}(\Sigma_i + \Sigma_j)|^{\frac{1}{2}}} + \sigma_\epsilon^2 \delta_{ij} \tag{7}$$

which is again a squared exponential[1]. It is easy to see that this new kernel function is a generalization of [3] by letting the covariance matrix of both inputs tends to zero. Hence, it is possible to learn from a combination of noise-free and uncertain inputs.

Theoretically, learning from uncertain data is as difficult as in the noise-free case, although it might require more data. The posterior distribution over function is found using the same equations by using the new covariance function. The hyperparameters can be learned with the log-likelihood as well, but it is now riddled with many local maxima. Using standard conjugate gradient methods will quickly lead to a local maxima that might not explain the data properly. An improper local maxima which occurs often is to interpret the observations as highly noisy. In this case, the matrix W tends to have

¹ In fact, the noise term is not a part of the integration since it models an independent noise process, and thus it remains in the new kernel.

large values on its diagonal, meaning that most dimensions are irrelevant, and the value of σ_ϵ^2 is over estimated to transpose the input error in the output dimensions.

A solution to prevent this difficulty is to find a maximum a posteriori (MAP) estimation of the hyperparameters. Placing a prior over the hyperparameters will thus act as a regularization term to prevent improper local maxima. In the experiments, we chose to use a prior of the exponential family in order to get a simpler log posterior function to maximize.

3 Experiments

In our experiments, we compare the performance of the Gaussian Process using inputs' uncertainty (noiseGP) and the standard Gaussian Process (classicGP) which use only the point estimates. We first evaluate the behavior of each method on a one-dimensional synthetic problem and then compare their performances on a harder problem which consists in learning the nonlinear dynamics of a cart-pole system.

3.1 Synthetic Problem: Sincsig

In order to be able to easily visualize the behavior of both GPs prior, we have chosen a one-dimensional function for the first learning example. The function is composed of a *sinc* and a *sigmoid* function as

$$y = \begin{cases} \text{sinc}(x) & \text{if } x \geq 0 \\ 0.5 [1 + \exp(-10x - 5)]^{-1} + 0.5 & \text{otherwise} \end{cases} \quad (8)$$

and we will refer to it as the *Sincsig* function. The evaluation has been conducted on randomly drawn training sets of different sizes. We uniformly sampled N inputs in $[-10, 10]$ which are the noise-free inputs $\{\mathbf{x}_i\}_{i=1}^N$. The observations set is then constructed by sampling each output according to $y_i \sim \mathcal{N}(\text{sincsig}(x_i), \sigma_y^2)$. The computation of the uncertain inputs is done by sampling the noise $\sigma_{x_i}^2$ to be applied on each input. For each noise-free x_i , we sampled the noisy input according to $u_i \sim \mathcal{N}(x_i, \sigma_{x_i}^2)$. It is easy to see that $x_i | u_i, \sigma_{x_i}^2 \sim \mathcal{N}(u_i, \sigma_{x_i}^2)$ and therefore we have a complete training set which is defined as $\mathcal{D} = \{(u_i, \sigma_{x_i}^2), y_i\}_{i=1}^N$. Figure 1 show a typical example of a training data set (crosses), with the real function to be regressed (solid line) and the result of the regression (thin line) for the noiseGP (top) and the classic GP (bottom). Error bars indicate that the classicGP is not consistent with the data since it does not take into account the noise's variance on inputs.

The first experiment was conducted with an output noise standard deviation $\sigma_y = 0.1$ with different size of training sets. The input noises standard deviation σ_{x_i} were sampled uniformly in $[0.5, 2.5]$. We chose these standard deviations so that adding artificially some independent noise during the optimisation process over the outputs can explain the noise over the inputs. All comparisons of the *classicGP* and the *noiseGP* has been done by training both with the same random data sets². Figure 2(a) shows the averaged mean square error over 25 randomly chosen training sets for different values

² Note that the standard Gaussian Process regression does not use the variances of the inputs.

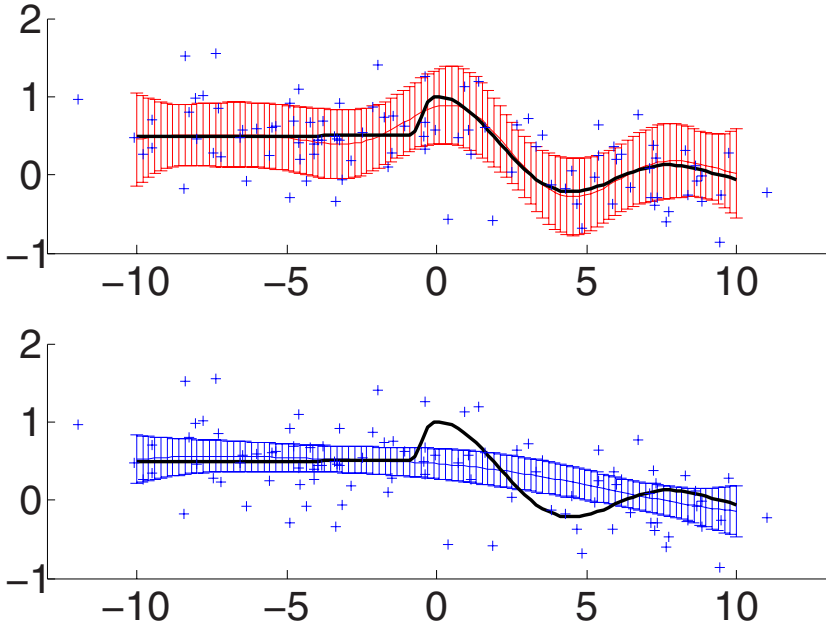


Fig. 1. The Sincsig function with classicGP and noiseGP regressions

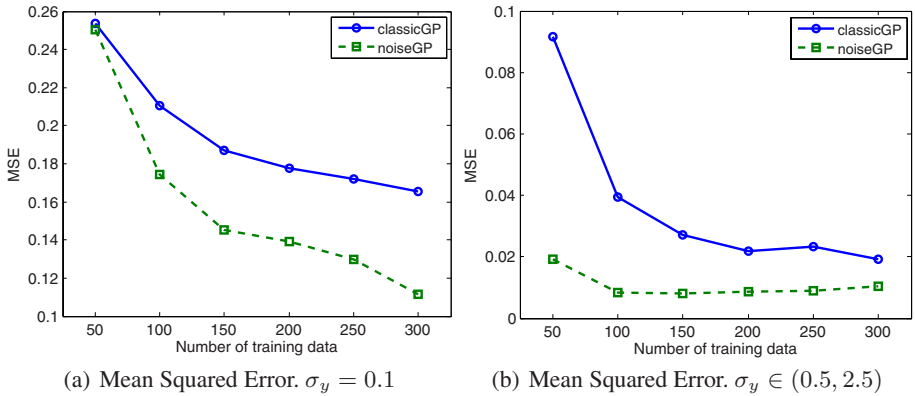


Fig. 2. Results on the Sincsig Problem

of N . Results show that when very few data are available, both processes explain the outputs with lot of noise over the outputs. As expected, when the size of the data set increases, the classicGP optimized its hyperparameters so as to explain the noisy inputs by very noisy outputs while the noiseGP correctly explain the noise on the inputs and selects the less noisy so as to minimize the mean squared error.

In the second experiment, in order to emphasize the impact of noisy inputs, we assumed that the Gaussian processes now know the noise’s variance on the observations. Therefore, the noise hyperparameters σ_ϵ^2 is set to zero since the processes exactly know the noise matrix to be added when computing the covariance matrix. For each output, the standard deviation σ_{y_i} is then uniformly sampled in $[0.2, 0.5]$. Figure 2(b) shows the performance of classicGP and noiseGP. Not allowing to explain noisy data by the independent noise process has two effects: First, it does not allow the classicGP to explain noisy inputs by noisy outputs when only few data are available, and it also forces the noiseGP to use the information on input variance whatever the size of the data set is.

Let us now see what the results on a real nonlinear dynamical system.

3.2 The Cart Pole Problem

We now consider the harder problem of learning the cart pole dynamics. Figure 3 gives a picture of the system from which we try to learn the dynamics. The state is defined by the position (φ) of the cart, its velocity ($\dot{\varphi}$), the pole’s angle (α) and its angular velocity ($\dot{\alpha}$). There is also a control input which is used to apply lateral forces on the cart. Following the equation in [9] to govern the dynamics, we used Euler’s method to update the system’s state:

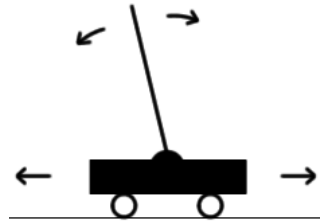


Fig. 3. The cart-pole balancing problem

$$\ddot{\alpha} = \frac{g \sin \alpha + \cos \alpha \left(\frac{-F - m_p l \dot{\alpha}^2 \sin \alpha}{m_c + m_p} \right)}{l \left(\frac{4}{3} - \frac{m_p \cos^2 \alpha}{m_c + m_p} \right)}$$

$$\ddot{\varphi} = \frac{F + m_p l (\dot{\alpha}^2 \sin \alpha - \ddot{\alpha} \sin \alpha)}{m_c + m_p}$$

Where g is the gravity force, F the force associated to the action, l the half-length of the cart, m_p the mass of the pole and m_c the mass of the cart.

For this problem, the training set were sampled exactly as in the Sincsig case. State-action pairs were uniformly sampled on their respective domains. The outputs were obtained with the true dynamical system and then perturbed with sampled noises assumed known. Since the output variances are also known, the training set can be seen as Gaussian input distributions that map to Gaussian output distributions. Therefore, one might use a sequence of Gaussian belief state as its training set in order to learn a partially observable dynamical system. Following this idea, there is no reason for the output distributions to have a significantly smaller variance than the input distribution.

In this experiment, the input and output noises standard deviation were uniformly sampled in $[0.5, 2.5]$ for each dimensions. Every output dimensions were treated independently by using a Gaussian Process prior for each of them. Figure 4 shows the averaged mean square error over 25 randomly chosen training sets for different N values for each dimension.

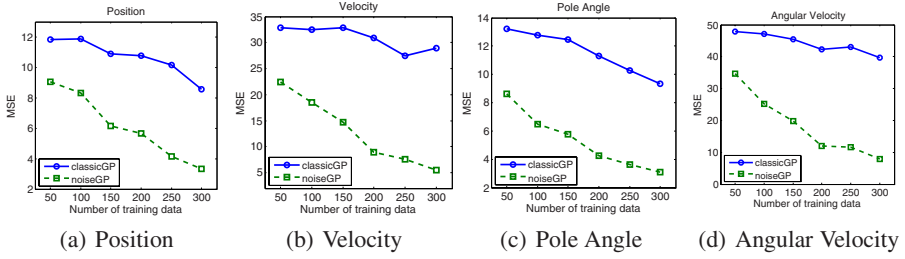


Fig. 4. Mean Squared Error results on the Cart-pole problem

3.3 Learning the Kernel Hyperparameters

As stated at the end of Section 2.2, it is possible to learn the hyperparameters given a training set. Since conjugate gradient methods performed poorly for the optimization of the log likelihood in the noiseGP cases, we preferred stochastic optimization methods for this task. In every experiments, we thus maximized the log posterior instead of the log likelihood. A gamma $\Gamma(2, 1)$ prior distributions as been placed over all characteristic length-scale terms in W and a normal $\mathcal{N}(0, 1)$ prior distribution as been placed over the signal standard deviation σ_f .

Comparing to previous work on the subject [6, 10] which use isotropic hyperparameters in the kernel function, we applied automatic relevance determination, that improves considerably the performance while does not increase the complexity of the kernel.

4 Discussion

Results for the synthetic problem are presented in Figure 1, 2(a) and 2(b). These results first show that using the knowledge of the noise on the inputs improve the consistency of the regression more than the standard Gaussian Process since the error assumed by the noiseGP includes completely the real function while the one of classicGP does not. Second, the noiseGP is also able to discriminate which noise comes from the input and which one come from the output as denoted in Figure 2(a) and 2(b). As the classicGP does not assume any noise on the input, it always assumes that the noise comes from the outputs, and thus learns a large hyperparameter for the noise, that also augments its mean squared error.

Problems of this approach come as soon as an optimisation of the hyperparameters have to be done. Indeed, the log-likelihood function is riddled of local maxima that cannot be avoided using classic gradient methods. An interesting avenue would be to look at natural gradient approaches [11]. Another future work concerns the application of this work to the learning of continuous Hidden Markov Models, as well as continuous POMDPs by using the belief state as a noisy input [12].

To conclude, we proposed a Gaussian Process Model for regression that is able to learn with noise on the inputs and on the outputs as well as to predict with less mean squared error than permitted by previous approaches while keeping consistent with the

true function. Results on a synthetic problem explain the advantages of the methods while results on the cart-pole problem show the applicability of the approach to the learning of real nonlinear dynamical systems, largely outperforming previous methods.

References

1. Golub, G., Loan, C.V.: An Analysis of the Total Least Squares problem. *SIAM J. Numer. Anal.* 17, 883–893 (1980)
2. Carroll, R., Ruppert, D., Stefanski, L.: *Measurement Error in Nonlinear Models*. Chapman and Hall, Boca Raton (1995)
3. Ghahramani, Z., Jordan, M.I.: Supervised learning from incomplete data via an EM approach. In: *NIPS*, pp. 120–127 (1993)
4. Tresp, V., Ahmad, S., Neuneier, R.: Training Neural Networks with Deficient Data. In: *NIPS*, pp. 128–135 (1993)
5. Quiñero-Candela, J., Roweis, S.T.: Data imputation and robust training with gaussian processes (2003)
6. Girard, A.: *Approximate Methods for Propagation of Uncertainty with Gaussian Process Model*. PhD thesis, University of Glasgow, Glasgow, UK (2004)
7. Rasmussen, C.E., Williams, C.K.I.: *Gaussian Processes for Machine Learning*. MIT Press, Cambridge (2006)
8. Girard, A., Rasmussen, C.E., Quiñero-Candela, J., Murray-Smith, R.: Gaussian Process Priors with Uncertain Inputs - Application to Multiple-Step Ahead Time Series Forecasting. In: *NIPS*, pp. 529–536 (2002)
9. Florian, R.: *Correct Equations for the Dynamics of the Cart-pole System*. Technical report, Center for Cognitive and Neural Studies (2007)
10. Quiñero-Candela, J.: *Learning with Uncertainty - Gaussian Processes and Relevance Vector Machines*. PhD thesis, Technical University of Denmark, Denmark (2004)
11. Roux, N.L., Manzagol, P.A., Bengio, Y.: Topmoumoute Online Natural Gradient Algorithm. In: *NIPS*, pp. 849–856 (2008)
12. Dallaire, P., Besse, C., Chaib-draa, B.: GP-POMDP: Bayesian Reinforcement Learning in Continuous POMDPs with Gaussian Processes. In: *Proc. of IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems (to appear, 2009)*

A Bootstrap Artificial Neural Network Based Heterogeneous Panel Unit Root Test in Case of Cross Sectional Independence

Christian de Peretti^{1,2,*}, Carole Siani³, and Mario Cerrato⁴

¹ Laboratory of Actuarial and Financial Sciences (SAF, EA2429), Institute of Financial and Insurance Sciences School, University of Lyon, France

² DEFI, EA4265, Department of Economics, University of Aix-Marseille 2, France
`Christian.de-Peretti@univmed.fr`

³ Research Laboratory in Knowledge Engineering (ERIC, EA3083), Department of Computer Science, University of Lyon, France
`carole.siani@univ-lyon1.fr`

⁴ Department of Economics, University of Glasgow, UK
`m.cerrato@lbss.gla.ac.uk`

Abstract. This paper extends an inference test proposed in [1]. The seminal paper proposes an artificial neural network (ANN) based panel unit root test in a dynamic heterogeneous panel context. The ANN is not complex, but it is not necessarily in the aim of modeling macroeconomic time series. However, it is applied in a difficult mathematical context, in which the classical Gaussian asymptotic probabilistic theory does not apply. Some asymptotic properties for the test were set, however, the small sample properties are not satisfactory. Consequently, in this paper, we propose to use the simulation based numerical method named “bootstrap” to compute the small sample distribution of the test statistics. An application to a panel of bilateral real exchange rate series with the US Dollar from the 20 major OECD countries is provided.

Keywords: Artificial neural network, panel unit root test, bootstrap, exchange rates.

1 Introduction

The standard linear autoregressive (AR) framework used to test for unit roots [2] in time series is increasingly viewed to be unsatisfactory and, as a result, alternative frameworks within which to test for unit roots are considered. For example,

* Corresponding author.

¹ A stochastic process is said to have a “unit root” if the polynomial corresponding to its AR structure has at least one of its roots equal to unity. This implies that the process is non-stationary, that has special meanings in economics: in particular the process is not mean-reverting, and thus the corresponding economic variable does not go back to the equilibrium (stable regime) in case of shock.

one alternative focuses on the use of panel data² and its role in improving the power of standard unit root tests. A good example is provided in [2], which uses a panel data test to reject the joint hypothesis of unit roots in each of a group real exchange rates against an alternative that they are all stationary. [3] provides a lucid general econometric discussion of panel methods. Another possible alternative is to allow different forms of stationarity to simple autoregressive moving average (ARMA) models. These include fractional integration (see [4]) and nonlinear transition dynamics (see [5]).

On one hand, using the likelihood framework, [6] proposes a testing procedure based on averaging individual unit root test statistics for panels. This test is referred to as the IPS test. On another hand, in [7] paper, the authors extend work on testing for unit roots against particular nonlinear alternatives by [8,9,10]. The resulting testing framework has power against a wide variety of nonlinear alternatives, which they established by using a Monte Carlo study. [1] proposes a panel unit root test that can now be based on the averages of the individual artificial neural network based Augmented DF (ADF) statistics proposed by [7]. In this paper, we propose bootstrap version of this test. The “bootstrap” we propose to use is a simulation based numerical method named to compute the small sample distribution of the test statistics.

The plan of the paper is as follows. Section 2 sets out the framework for neural networks, and derives neural test statistics in the case where errors in individual Dickey-Fuller regressions are serially uncorrelated. The cases of trended series is also discussed. Section 3 presents the bootstrap procedures. An application to a panel of bilateral real exchange rate series with the US Dollar from the 20 major OECD countries is provided in Sect. 4. Finally, Sect. 5 provides some concluding remarks.

2 Neural Unit Root Tests for Heterogeneous Panels

In this section, the test based on artificial neural networks of [1] is presented.

2.1 The Basic Framework

Consider a sample of N cross sections (industries, regions or countries) observed over T time periods. Suppose that the stochastic process, $(y_{it})_{it}$, is generated by the nonlinear first-order autoregressive process:

$$y_{it} = \tilde{f}_i(y_{i,t-1}) + \varepsilon_{it}, \quad i = 1, \dots, N; \quad t = 1, \dots, T, \quad (1)$$

² Panel data are two dimensions datasets: one time dimension, and one individual (firms, countries, stocks, ...) dimension. The panel models used to handle these data are very difficult to treat: they have to account for the classical problem encountered in individual models, the autocorrelation problems in time series models, plus additional cross section problems. In addition, in the case of non-stationary panel models, there are almost no mathematical properties that are proved formally.

where initial values, y_{i0} , are given. It should be noted that this panel is heterogeneous: this formulation allows for \tilde{f}_i to differ across groups. We are interested in testing the null hypothesis of unit roots

$$P \left\{ \tilde{f}_i (y_{i,t-1}) = y_{i,t-1} \right\} = 1$$

for all i . Model 1 can be expressed as

$$\Delta y_{it} = f_i (y_{i,t-1}) + \varepsilon_{it}, \tag{2}$$

where $f_i (y_{i,t-1}) = \tilde{f}_i (y_{i,t-1}) - y_{i,t-1}$. The null hypothesis of unit roots then becomes

$$H_0 : P \{ f_i (y_{i,t-1}) = 0 \} = 1 \text{ for all } i, \tag{3a}$$

against the alternatives

$$H_1 : \begin{cases} P \{ f_i (y_{i,t-1}) = 0 \} < 1, & i = 1, 2, \dots, N_1, \\ P \{ f_i (y_{i,t-1}) = 0 \} = 1, & i = N_1 + 1, N_1 + 2, \dots, N. \end{cases} \tag{3b}$$

This formulation of the alternative hypothesis allows for some (but not all) of the individual series to have unit roots under the alternative hypothesis³

In this context, the great advantage of artificial neural networks arises out of their potential to approximate arbitrary nonlinear functions. The general form of an artificial neural network approximation applied in this context is given by

$$\Delta y_{it} = \alpha_i + \sum_{k=1}^{q_i} \beta_{ik} \psi_{ik}(x_{it}) + \varepsilon_{it}, \tag{4}$$

where q_i is the number of ‘hidden nodes’ or ‘units’, $(\beta_{ik})_k$ the unknown parameters, $(\psi_{ik})_k$ some scalar functions, and $(x_{it})_t$ the vectors of inputs. Work of [11,12] suggests that radial basis functions may provide more powerful artificial neural network tests in a number of circumstances. A radial basis function is a function which is monotonic about some centers, $\psi_{ik}(x_{it}) = \psi(x_{it}; c_{ik}, r_i)$. In this case, $x_{it} = y_{i,t-1}$. A test for unit root can be provided by imposing the following in Model 4⁴

$$\beta_{i1} = \beta_{i2} = \dots = \beta_{iq_i} = 0 \tag{5}$$

³ Formally, following [6], we assume the following assertion: under the alternative hypothesis the fraction of the individual processes that are stationary is non-zero, namely $\lim_{N \rightarrow \infty} \frac{N_1}{N} = \delta, 0 < \delta \leq 1$.

⁴ The treatment of deterministic terms such as a constant and a trend are handled straightforwardly by demeaning or demeaning and detrending the series prior to applying the test. Nevertheless, the constant term α_i has to be kept in Model 4 since the neural component is a nonlinear function of the inputs, and is not necessarily zero-mean.

2.2 Neural Unit Root Tests for Heterogeneous Panels

In this section, the artificial neural network based test in the context of the panel data model [1](#) is recalled. The purpose is to test the hypothesis that the series are random walks against the alternative of a proportion of them are being stationary ergodic linear or nonlinear processes. The error terms are assumed to be serially uncorrelated. For this purpose the following assumption is made:

Assumption 1:

ε_{it} , $i = 1, \dots, N$, $t = 1, \dots, T$, in Model [1](#) are independently and normally distributed random variables for all i and t , with zero means and finite heterogeneous variances σ_i^2 .

In this case the relevant Radial Basis Function (RBF) regressions are given by Model [4](#). Extending [7](#) to the panel context, an artificial neural network based test can be developed as follows. Define q_i centers c_{ik} , $k = 1, \dots, q_i$, and a radius r_i . The Gaussian RBF is

$$\psi(y_{i,t-1}; c_{ik}, r_i) = \exp \left(- \left(\frac{y_{i,t-1} - c_{ik}}{r_i} \right)^2 \right), \tag{6}$$

where $\| \cdot \|$ denotes a norm. See [13](#) for a nontechnical introduction to artificial neural networks in general, which covers RBF networks. [14](#) gives a more thorough account.

The center vectors $(c_{ik})_k$, the radius vector r_i , and the number of hidden units used have to be determined. Data-based procedures are used for all (they can be provided under request to the corresponding author). The RBF test used is a standard Wald test to test the null hypothesis that $\beta_{i1} = \beta_{i2} = \dots = \beta_{iq_i} = 0$. This takes the form:

$$t_i^*(T, q_i) = \frac{1}{\hat{\sigma}_i^2} \hat{\beta}'_i [R'(W'_i W_i)^{-1} R]^{-1} \hat{\beta}_i, \tag{7}$$

where $\hat{\beta}'_i = (\hat{\beta}_{i1}, \hat{\beta}_{i2}, \dots, \hat{\beta}_{iq_i})'$ is the estimated parameter vector, $\hat{\sigma}_i^2$ is the estimated variance of the residuals, R is the selector matrix, W_i is the matrix of regressors of Model [4](#) including the constant. In the case of stationary series, $t_i^*(T)$ is asymptotically distributed $\chi^2_{q_i}$ under the null.

The major problem with the unit root tests is that the test statistics do not have a standard student or χ^2 distribution under the null hypothesis of a unit root. For instance, the Dickey-Fuller t -statistic does not have a student distribution but a Dickey-Fuller distribution. In the case of the individual RBF test, [7](#) proposes to use the bootstrap technique. Figure [1](#) displays the probability density function of standardized $t_i^*(T, q_i)$ statistic for $T = 200$, $q_i = 2$, and 10,000 Monte Carlo replications [5](#).

⁵ It should be noted that even if bootstrap techniques are used, it should be checked that at least the two first moments of the distribution exist. The two first moments were examined in [11](#). However, for small values of T , the use of asymptotic moment values could lead to poor test results.

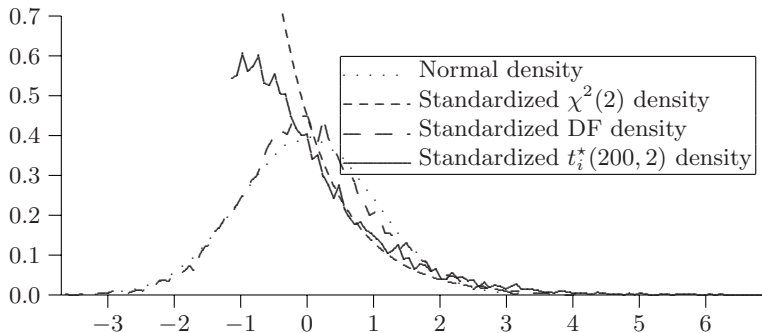


Fig. 1. Probability density Function of the standardized $t_i^*(200, 2)$ statistic

Here, in the case of panel, we are not interested in the individual RBF test statistic distribution (bootstrapped or not), but in constructing a panel RBF test. Here we follow [6] and focus on a panel unit root test based on the average of individual statistics. Therefore, for a fixed T (the focus of this section), the following average statistic is considered:

$$\bar{t}^*(N, T) = \frac{1}{N} \sum_{i=1}^N t_i^*(T, q_i). \tag{8}$$

2.3 Case of Fixed N and T

When N , T_i and q_i are fixed, the sample distributions of the test statistics, under the null hypothesis [5] are non-standard, but do not depend on any nuisance parameters. Exact sample critical values for the statistics in this case could be computed via stochastic simulation. However, the distributions depend on N , T_i and q_i and it is not reasonable to provide such results, even if $T_1 = \dots = T_N = T$ and $q_1 = \dots = q_N = q$. It can be observed in [1] that the skewness of the panel test statistic is still not negligible for values of N and T encountered in practice (for instance, for $N = 20$ and $T = 100$, the skewness is between 0.3 and 0.5, depending on q , and its kurtosis is between 3.13 and 3.26). Bootstrap techniques can also be used as in [7]. Consequently, a bootstrap procedure for using this neural test can be developed.

3 Bootstrap Test

In this section, we propose a bootstrap version of the neural test in the context of the panel data model defined in [1]. The purpose is to test the hypothesis that the series are all non-stationary (in the sense of random walks) against the alternative of a proportion of them are being stationary ergodic linear or nonlinear processes (see [3a] and [3b]).

We suggest a bootstrap procedure for the following reasons:

1. The normal approximation is not necessarily satisfactory enough for samples sizes encountered in practice (see previous subsection).
2. The optimal number of hidden nodes estimated on the real data (determined by a criterion as the AIC) is not necessarily realistic with the null hypothesis that should have a smaller number of nodes, and then provide a smaller t^* statistic. By allowing our bootstrap procedure to re-assess what would be the optimal number of hidden nodes under the null, that permits a better discrimination between the null hypothesis, and the alternative hypothesis that provides a greater test statistic.
3. Finally, since in the case of cross sectional dependence the test statistic is not asymptotically normal, the bootstrap procedure is an alternative to provide numbers of tables of critical values with respect to N and T .

3.1 Bootstrap Procedure

1. The test statistic is computed on the original sample of panel data. Let denote it τ . The optimal number of hidden nodes is assessed with the AIC criterion for each individual time series in the panel.
2. The original data are estimated under the null hypothesis, that is an independent panel of unit root processes. Consequently, the standard deviations for each series are estimated using simply the empirical standard error estimator.
3. B simulated samples are generated following the estimated Data Generating Process (DGP) under the null: a panel of independent unit root processes with variances given by the estimates in step 2.⁶ The bootstrap can be parametric (for instance a Gaussian panel), or nonparametric (using resampling in the vector of residuals)⁷
4. For each simulated sample, the test statistic is computed. Let denote it τ_b . The important point is that the optimal number of hidden nodes found in step 1 is not imposed. The optimal number of hidden nodes is re-assessed using the AIC for each new simulated sample.
5. The bootstrap P value is then computed as follows: $\frac{1}{B} \sum_{b=1}^B I(\tau_b > \tau)$.

4 Application to a Panel of Bilateral Real Exchange Rates

In this section, our test is applied to real exchange rates against the US Dollar for twenty OECD countries over the period 1973Q1–1998Q2. The data set is the same used by [\[16,17\]](#).

⁶ For generating individual unit root processes under the null, see [\[15\]](#).

⁷ Given the universal function approximation ability of artificial neural networks, the sequence of residuals should converge to an i.i.d. sequence asymptotically if the number of hidden units is allowed to grow with the sample size, both under the null hypothesis and under the alternative hypothesis.

Table 1. P values of the individual Unit Root Tests for Real Dollar Exchange Rates

Country	p^\dagger	ADF	BK \dagger	q^\dagger	BK \ddagger	q^\ddagger
Australia	0	0.6076	0.6146	3		
Austria	0	0.3153	0.5185	1		
Belgium	2	0.5155	0.5656	2		
Canada	3	0.7427	0.8539	2		
Denmark	2	0.4525	0.1091	3		
Finland	3	0.0891	0.1341	2		
France	0	0.3704	0.1471	3		
Germany	2	0.4024	0.6937	1		
Greece	4	0.1752	0.2733	2		
Ireland	2	0.2472	0.4525	2		
Italy	0	0.3483	0.2893	2		
Japan	0	0.4474	0.6086	2		
Netherlands	2	0.4254	0.6066	1		
N Zealand	0	0.3333	0.3213	2		
Norway	2	0.4254	0.1421	2	0.0090	*** 3
Portugal	0	0.5265	0.5566	2		
Spain	1	0.3223	0.4164	2		
Sweden	3	0.2192	0.0020	*** 3		
Switzerland	1	0.2583	0.0140	** 3		
UK	2	0.2803	0.3794	3		

\dagger p is the number of lagged regressors,

BK denotes the [7] test P value.

q is the number of neural regressors.

\ddagger The BK test is rerun with one additional neural regressor (in case of the AIC under-estimates the number of neural regressors).

* P value significant at 2% level,

** P value significant at 1% level.

Since the long run *Purchasing Power Parity* (PPP) relationship is one of the main components of theoretical international macroeconomic models, a large number of studies have tested this relationship by applying unit root tests to real exchange rates. Most of these studies show evidence of unit root behavior in real exchange rates, which has become a puzzle in international finance. The growing literature on nonlinear exchange rates argues that transaction costs and frictions in financial markets may lead to nonlinear convergence in real exchange rates. Consequently, the non-mean reversion reported by linear unit root tests may be due to the fact that these tests are based on a mis-specified stochastic process.

4.1 Univariate Unit Root Tests

First, the ADF test and the [7] test are applied to the separate individual time series of exchange rates. An intercept, but no trend is included. The statistics are bootstrapped with $B = 999$ bootstrap replications. The results are presented in Tab. [1].

Table 2. P values for the panel Unit Root Tests for Real Dollar Exchange Rates

Test	IPS	ANN
Asymptotic	0.0413 *	0.0248 *
Bootstrap	0.0486 *	0.0010 ***

* P value significant at 5% level,

*** P value significant at 1% level.

ADF test rejects the unit root null hypothesis in 0 out of 20 cases at all levels of significance. By contrast, the [7] test rejects the null in 3 cases at the 2% significance level. Hence the [7] test rejects the unit root null more frequently and therefore yields stronger support for the long-run PPP.

4.2 Panel Unit Root Tests

As we argued above, univariate tests have low power and this problem is overcome by employing panel unit root tests. The results for the IPS test and for our neural panel unit root test are shown in Tab. [2]. The number of lags used in these tests are the same as for individual tests. Three neural regressors are used for each series in the neural panel test. The bootstrap test is run with $B = 9999$.

The contrast between the two bootstrap panel statistics is rather strong. IPS test fails to reject clearly the unit root null at all levels of significance: the P values (asymptotic and bootstrap) are close to the 5% level limit, and thus may imply non-mean reversion in the whole panel of real exchange rates. On the other hand, our bootstrap neural panel test rejects the null hypothesis of unit root for the panel of real exchange rates at all levels of significance, giving support to the long-run PPP for the whole panel of OECD countries. This evidence of nonlinear mean reversion in the OECD real exchange rates may suggest that previous evidence in the literature of non-mean reversion in real exchange rates is due to using linear unit root tests. The evidence we provide is more in accordance with what is expected by the economic theory.

5 Concluding Remarks

The focus of econometric investigation has shifted away from stable linear and unit root processes towards more general classes of processes that include nonlinear specifications. Some tools to distinguish the nature of empirical series were developed, as in [9] and [7]. However, the only paper proposing a unit root test in a nonlinear panel framework is [18], which proposes a test against a very specific nonlinear alternative. In this paper, we have developed a bootstrap procedure to improve the artificial neural network test of [11] for testing the unit root hypothesis against stable nonlinear processes in heterogeneous panels. In the case of panel framework, it is possible to substantially augment the power

of the unit root tests applied to single time series. Our application to bilateral real exchange rates shows that the empirical results can be very different, and our test permits to display evidence expected by economists that classical tests cannot.

Acknowledgments. The authors wish to thank the referee of the 16th *International Conference on Neural Information Processing* for its useful comments.

References

1. de Peretti, C., Siani, C., Cerrato, M.: An artificial neural network based heterogeneous panel unit root test in case of cross sectional independence. In: 2009 International Joint Conference on Neural Network (IJCNN 2009), Atlanta, June 14-19. International Neural Network Society, IEEE Computational Intelligence Society (Forthcoming, 2009)
2. Abuaf, N., Jorion, P.: Purchasing power parity in the long run. *Journal of Finance* 45, 157–174 (1990)
3. Im, K., Pesaran, M.H., Shin, Y.: Testing for unit roots in heterogeneous panels. Unpublished manuscript, University of Cambridge (1997), <http://www.econ.cam.ac.uk/faculty/pesaran>
4. Mills, T.: *The Econometric Modelling of Financial Time Series*. Cambridge University Press, Cambridge (1993)
5. Pesaran, M.H., Potter, S.: A floor and ceiling model of us output. *Journal of Economic Dynamics and Control* 21(4/5), 661–696 (1997)
6. Im, K., Pesaran, H., Shin, Y.: Testing for unit roots in heterogeneous panels. *Journal of Econometrics* 115, 53–74 (2003)
7. Blake, A.P., Kapetanios, G.: Pure significance tests of the unit root hypothesis against nonlinear alternatives. *Journal of Time Series Analysis* 24(3), 253–267 (2003)
8. Caner, M., Hansen, B.: Threshold autoregression with a near unit root. Unpublished manuscript, University of Wisconsin (2000)
9. Kapetanios, G., Shin, Y., Snell, A.: Testing for a unit root in the nonlinear star framework. *Journal of Econometrics* 112, 359–379 (2003)
10. Kapetanios, G., Shin, Y.: Testing for a unit root against threshold nonlinearity. Unpublished manuscript, University of Edinburgh (2000)
11. Blake, A.P., Kapetanios, G.: A radial basis function artificial neural network test for arch. *Economics Letters* 69(1), 15–25 (2000)
12. Blake, A.P., Kapetanios, G.: A radial basis function artificial neural network test for neglected nonlinearity. Working Paper, National Institute of Economic and Social Research (2000)
13. Campbell, J.Y., Lo, A.W., MacKinlay, A.C.: *The Econometrics of Financial Markets*. Princeton University Press, Princeton (1997)
14. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford (1995)
15. Basawa, I.V., Mallick, A.K., McCormick, W.P., Reeves, J.H., Taylor, R.L.: Bootstrapping unstable first order autoregressive processes. *Annals of Statistics* 19(2), 1098–1101 (1991)

16. Murray, C.J., Papell, D.H.: The purchasing power persistence paradigm. *Journal of International Economics* 56, 1–19 (2002)
17. Murray, C.J., Papell, D.H.: The purchasing power puzzle is worse than you think. *Empirical Economics* 30(3), 783–790 (2005)
18. Cerrato, M., de Peretti, C., Sarantis, N.: A nonlinear panel unit root test under cross section dependence. Working paper. Centre for International Capital Markets, London Metropolitan Business School, London Metropolitan University (2007)

A Novel Hierarchical Constructive BackPropagation with Memory for Teaching a Robot the Names of Things

Fady Alnajjar¹, Abdul Rahman Hafiz², and Kazuyuki Murase^{1,2,3}

¹ Department of System Design Engineering, IEEE student member

² Department of Humana and Artificial Intelligent System

³ Research and Education Program for Life Science, Graduate School of Engineering,
University of Fukui, Japan

{fady, abdul, murase}@synapse.his.fukui-u.ac.jp

Abstract. In recent years, there has been a growing attention to develop a Human-like Robot controller that hopes to move the robots closer to face real world applications. Several approaches have been proposed to support the learning phase in such a controller, such as learning through observation and/or a direct guidance from the user. These approaches, however, require incremental learning and memorizing techniques, where the robot can design its internal system and keep retraining it overtime. This study, therefore, investigates a new idea to develop incremental learning and memory model, we called, a Hierarchical Constructive BackPropagation with Memory (HCBPM). The validity of the model was tested in teaching a robot a group of names (colors). The experimental results indicate the efficiency of the model to build a social learning environment between the user and the robot. The robot could learn various color names and its different phases, and retrieve these data easily to teach another user what it had learned.

Keywords: Incremental learning and memory, human-like robot controller, constructive backpropagation.

1 Introduction

Developing a complete human-like robot controller, inspired from the principles of neuroscience, is one of the challenging tasks for many groups of robotics researchers [1]. The difficulty in such a system can be summarized in three main points as diagrammatically shown in Fig.1: i) A simple mechanism for human-robot interaction, which is, mainly relies on robot's vision, speech recognition, sensor-motor interaction, etc. ii) A dynamic mechanism for learning and memory, which gives the robot the features to learn and/or to teach. iii) A mechanism for homeostatic, which gives the robot a degree of an internal stability.

In our early works [2], we have proposed a model for better robot's vision toward better human-robot interaction (level-1 in Fig.1). Following this series of study, in this paper, we are highlighting the issue of enhancing the robot's learning and memory (level-2 in Fig.1).

In general, robot can learn behaviors either by: i) independent learning, i.e. without the needs to interact with human, such as simple obstacle avoidance, or target tracking,

etc., where the learning can be autonomously occurred by the known unsupervised evolutionary or adaptation algorithms (genetic algorithms, Hebbian learning, etc.) [3][4][5]. Or by ii) Un-independent learning, such as learning a particular skill or learning the names of various things, in which the interaction with human is needed. Such supervised learning can be achieved either by observing mechanisms, where the robot can observe the human actions and mapping it into its behavior [6], or by a direct guidance from the human, where the user can take a walk with the robot in a room and keeps teaching the robot the names of things around in natural way (exactly as he teaches his child).

To support such as un-independent learning, incremental learning and memory structure can be the most suitable structure so far, since its size is adaptable to the amount of data that the robot may learn during its life. These data are usually dynamic and unpredictable. Giving a static structure to such a system could potentially run into problems like under fitting, over fitting or even wasting computational resources.

In this study, therefore, we are suggesting an incremental learning and memory model, where a new skill or object names can be easily taught to the robot. We called the model a Hierarchical Constructive BackPropagation with Memory (HCBPM). The validity of the model is tested in a task of teaching a human-like robot “RoboVie-R2” the names of various things (colors for simplicity) by a normal user. Some image processing and sound recognition algorithms are borrowed from our early works to support the system [2]. The experimental results shows that the robot could learn color names and its different phases, and could retrieve these data easily from its memory.

The following section highlights a brief history of incremental learning and memory. The rest of the paper is organized as follows. Section 3, describes in details the proposed model. Section 4, represents the robot and the task. Section 5, shows experimental setup and results. Finally section 6, concludes the work and points at possible future research directions.

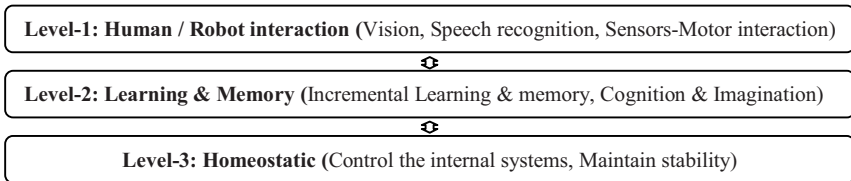


Fig. 1. Human-like robot’s controller

2 The History of Incremental Learning and Memory

So far, the mechanism of how human’s brain works to learn and memorize behaviors or names is a very complicated issue, and it is still a subject of hot debate for many groups of researchers [1]. Although the exact principles of such a mechanism are not yet clear, many researchers have agreed on some of its main features that can help to design a similar one to achieve a human-like robot controller. For example: i) The learning and memory techniques should fall somewhere in between stability and

plasticity spectrum. ii) Synaptic weights should code some knowledge of the past experiences. iii) Adaptable to dynamic changes with minimum computational time.

Although many algorithms have been developed for learning and memory to satisfy such a mechanism with various degrees of success [7] [8], we believe that the constructive backpropagation algorithm [9] [10], with some minor amendment, can be the key to step forward toward such a target. It supports incremental learning in real-time with reasonable computational time [10], and it has a degree of memory. This degree, however, is limited and might not form long-term memory in some domains, since it may be disturbed by additional learning of new data. Therefore, attaching a separate memory level that can guarantee the stability besides the plasticity of the system is required.

In this study, therefore, we are proposing a model that has ability to keep learning new information with its various phases without forgetting previously acquired knowledge and can retrieve this information easily. The model is presented by three-level *HCBPM*, with respect to all the features mentioned above. We believe that this model is an indispensable tool for teaching the robot in natural way.

3 A Hierarchical Constructive BackPropagation with Memory

This section describes the *HCBPM* model (Fig.2), and the working mechanism of each level. From the figure, *HCBPM* is represented by three levels: i) Network Switcher (NS), which is used to learn different phases of the object and to switch it to its original form, before passing it to the next level. ii) Constructive BackPropagation network (CBP), which is used for incremental learning. iii) Memory Space (MS), which is used for storing and retrieving the data.

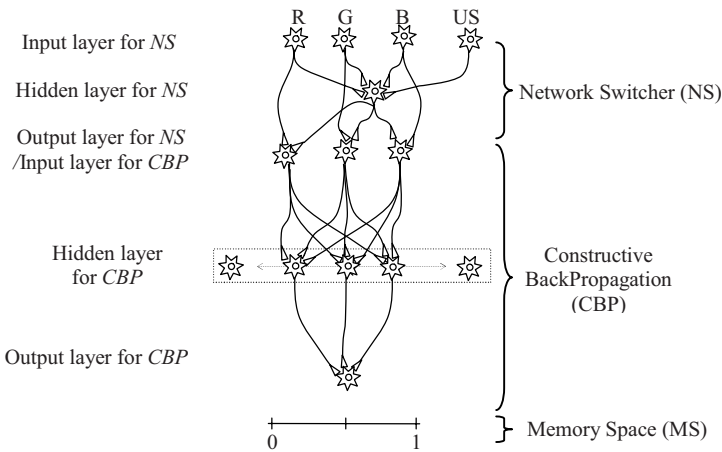


Fig. 2. A schematic model of the *HCBPM*

3.1 Network Switcher (NS)

NS represents the upper level of *HCBPM* (Fig.2). It is used to learn different phases of the object. It has three layers: i) Input-layer that has similar number of neurons to its

output, plus one addition neuron, called user sensor (US), which is used to confirm the input from the user before it activates and trains the network. The neurons in this layer are connected to both the hidden layer neurons and the output layer neurons, except for the *US* neuron, which is only connected to the hidden layer neuron. ii) Hidden layer that works as a switcher to the network. It has excitatory and/or inhibitory impact to the network output neurons. This layer is activated either by the user comment or by the amount of the inputs that could reach a certain threshold value assigned during the earlier training of this layer. If the input object to this network is in its original form, this layer will not be activated. iii) The output layer that represents the input neurons of the *CBP* network.

3.2 Constructive BackPropagation (CBP)

CBP is a three-layer network structure used by the robot to learn various names (Fig.2). The hidden-layer is initialized by two neurons and can be incrementally increased based on the amount of data that the robot can learn during its life. The output layer contains one neuron that maps the network output to the *MS* level. *CBP* is trained by the constructive back-propagation algorithm [9] [10] [11]. The flowchart in Fig.3, explains in details the working mechanism of *CBP*. In brief:

- Weights in *CBP* are randomly initialized.
- Robot reads the front object by its camera (since we are dealing with colors, the robot reads the RGB of the color, R: red, G: green, B: blue).
- If the robot has previously experienced the color, i.e. it is already mapped into its memory; the robot will identify the color, call it from its memory and say its name.
- If the robot has not experienced the color, it will ask the user about its name, assign a particular data point with a certain range for the color, and check the possibility of any overlap between the new data and the existed data in its memory.
- If there is an overlap, the error tolerance (ET) will be gradually decreased, so that, the range of each data point in the *MS* will be shrunken to open a new space for upcoming data points and then continue the training.
- During the training, if the error rate (ER) reaches to a value that is equal or less than the ET, then the training will be stopped and the learning will be confirmed. Otherwise, the memory space will be expanded by adding a hidden neuron to the *CBP* level, if and only if, the training does not reach to its target within 500 cycles.
- Continue the training, jump to step 6.

Although constructive learning algorithms have many advantages [10], they are very sensitive to changes in the stopping criteria. If training is too short, the components of the network will not work well to generate good results. If training is too long, it costs much computation time and may result in over fitting and poor generalization. Therefore, in this stage we have selected a variable stopping criteria *ET* that can be gradually decreased during the learning to satisfy the training requirement at that time.

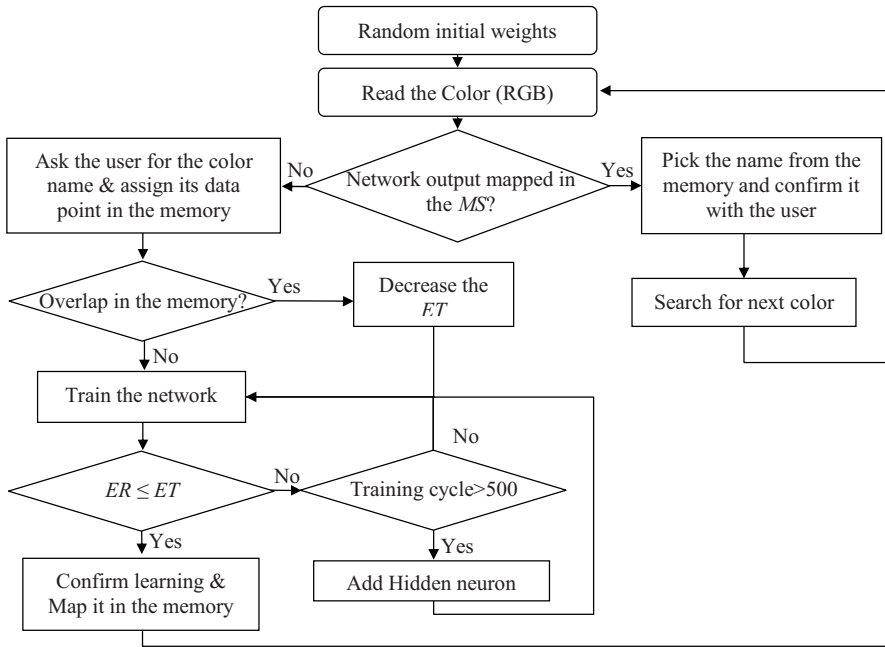


Fig. 3. Flowchart illustrates the *CBP*'s working mechanism

3.3 Memory Space (MS)

The *MS* in this level is represented by a number of data points that each of which represents a name of the color that the robot learns from the user. These points are assigned between 0 and 1. Each of these points has a range that is changeable based on the value of the variable *ET*, which indeed based on the density of the data in the memory. The number of the data points along the *MS* is assigned by the size of the hidden layer neurons in *CBP*'s level.

Assigning the data points in the *MS* for each upcoming object is done orderly by the *CBP*'s network output neuron in order to control the network training direction (more explanation is in experiment 5.1).

4 The Robot and the Task

All the works in this study have been conducted in a physical human-like robot (Robovie-R2) [12] (Fig.4A). Robovie-R2 equipped with various types of sensors and motors. In this study, the color camera and the microphone that was mounted in the robot's head were used. The camera was used to read the colors, and it was also used with the microphone to facilitate the interaction task with the user.

The robot task was to learn from a normal user a group of color names and its different phases, and to retrieve these data to teach another user what it had learned.

5 Validation of the Framework

This section presents the experimental validation of the work. In the following experiment, all the synaptic weights in the *HCBPM*'s level were initialized randomly and the *MS* were set up empty (Fig.5A).

5.1 Interacting with User (Learning and Memory)

In this experiment, a user showed sequentially four different colors to the robot (red, green, blue and pink) and asked the robot for their names. The following points illustrate the scenario that occurred during the experiment:

- The user first showed the robot the red color and asked “Do you know what this color is?”
- The robot looked at the color, took four samples, and read the RGB of each sample.
- The robot tested the samples by its network and found that it is a new color, which it had not experienced before. The robot, therefore, answered: “No, I don’t know, Can you please tell me what this color is?”
- The user answered the robot: “it is Red”
- The robot assigned a data point for the color in its *MS* based on (Eq.1), where in this case Red=0.5. The *CBP*'s level was then trained by *BP*, where the new samples of RGB represented the input training set of the network and (red=0.5) represented the desired output.
- After training the network and storing the new data in its memory, the robot confirmed the training “Thank you, I know now what is Red”.
- The user continued showing the rest of the colors to the robot and similar scenarios were occurred.

Figure 5, shows the steps of learning each of the color's name and storing it in the robot's *MS*. Notice that all the colors in this stage were in its original phase.

From the figure, *CBP* was initialized by two hidden neurons and the *ET* was set to 0.2. During the learning, *ET* was gradually decreased (Fig.6). This decrease shortened the range of each data point in the *MS* to open a new space for upcoming data. Two hidden neurons were sufficient to learn the first three colors (Fig.5B,C&D). When the user showed the forth color, *CBP*'s level failed to train the network to the target ($ER \leq ET$), therefore, a new hidden neuron was added into the *CBP*'s level. This additional neuron expanded the memory and gave new space. *CBP* therefore could continue the training.

After teaching the robot the four colors, the user reshaw the colors to the robot and the robot could identify each of these colors successfully.

$$a = (b + c) / 2 \tag{1}$$

Where b&c assigned consistently in the *MS* as shown in the Fig.5A~E.

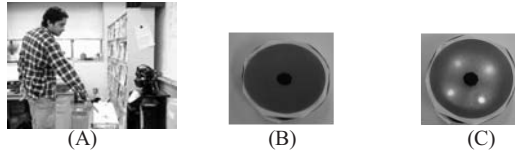


Fig. 4. (A) RoboVie-R2 while reading the color which given by the user. (B) A sample of the red color in its original form. (C) The red color in the Light-on phase.

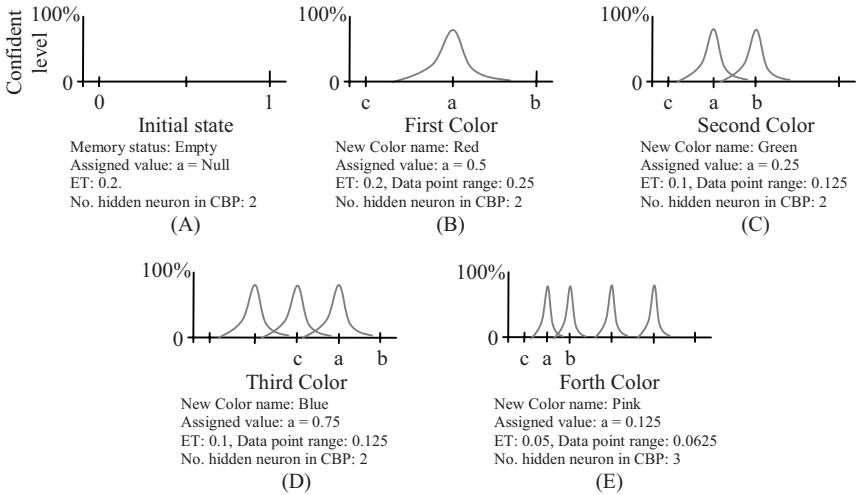


Fig. 5. The steps to store new color names in the memory space

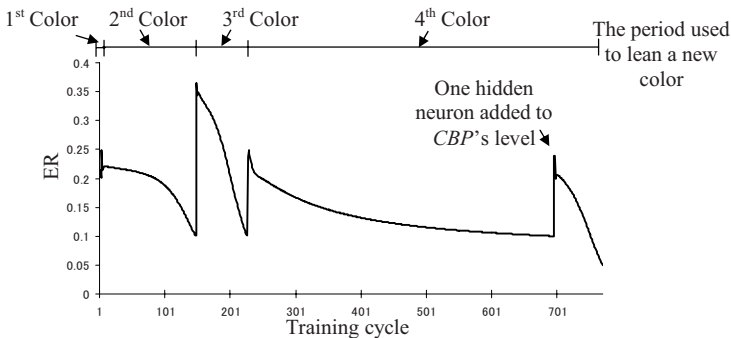


Fig. 6. ER during the learning. ET for the first color was set to 0.2, for the second and the third color were set to 0.1, and for the forth color was set to 0.05.

5.2 Interacting with User (Learning Different Phases)

This experiment is to examine the validity of NS's level to learn the different phases of each color and switch it to its original before it hands it over to the CBP's level.

This level requires the guidance from the user at the early stages to set up the threshold value of its hidden layer neuron.

In this level, the user showed the robot the red color again but with its new phase (Light-on) (Fig.4C). The robot read the samples of the new RGB, which is, for simplicity, a representation of a regular shifted form from its original. Since *NS* was not yet trained, the robot, therefore, assumed it as a new color and asked the user of its name as follows:

- Robot: “I don’t know, Can you please tell me what this color is?”.
- User: “This is Red but Light-on”.
- Since the original red color has been trained before, the robot activated the *NS*’s level and trained it by *BP*. The network input training sets were the new RGB samples and the network desired output was the nearest value of the original form of the red color.
- The hidden layer neuron in *NS*’s level was then assigned by a threshold value that can be activated by any other color with its “Light-on” phase.

To confirm the learning of the *NS*’s level, the user trained the robot with various samples of the red on its “light-on” phase. For the testing stage, the user, showed a green color with (light-on) to the robot. Interestingly, even that the robot had not experienced the green color with the light-on phase, the RGB of this phase could reach the threshold value and activate the *NS*’s level. The robot could successfully identify the color and its phase “This is Green Light-on”.

We believe that different phases of the original form of any color can be learned by similar scenario.

5.3 Interacting with User (Retrieving Existing Data to Teach Another User)

In this experiment, we examine the ability of the robot to retrieve the information that it learned from the above experiments to teach another user the names of the colors.

The scenario was similar to the one in the first experiment but with a replacing between the position of the user and the robot. In this experiment, the robot started to point randomly at the colors and asks the user about its names. If the user did not know the name, the robot taught him. This experiment was carried out successfully. Experiment results were omitted due to page limitation.

6 Conclusion

This paper addressed a new method for learning and memory, where a new object names and its different phases can be taught easily to the robot. This work can be considered as a social learning architecture, inspired from how the human can teach his child the names of things around. The framework is based on three-level hierarchical controller each of which responsible in a part of the work. The first level is the network switcher *NS*, which is used for learning different phases of the object to switch it to its original form before passing it to the next level. The second is the Constructive BackPropagation *CBP*, which is used for incremental learning. The third level is the memory space level *MS*, which is used for storing and retrieving the data that robot learned during its life.

The training, in the experiment section, took place in real-time and the architecture scaled from simple to nearly complex task. Experimental results indicate that the proposed model works rather well in practice and could develop a positive interaction between the robot and the user. The robot could learn new color names with its different phases, and retrieve the old data easily to teach another user what it had learned. We believe that the proposed model in this study is an indispensable tool for teaching the robot in natural way. This paper satisfies the learning and memory part of the human-like robot controller (Fig.1).

For future research, we intend to further examine the model in a wider range of office-like environment and higher complex task after improving the image processing part, where the robot can see and understand different objects (e.g., TV, video, PC, desk, etc.) and its different phases such as size, shape, etc.

Acknowledgments. This work was supported by grants to KM from Japanese Society for Promotion of Sciences and from the University of Fukui.

References

1. Floreano, D., Mattiussi, C.: *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*. The MIT Press, Cambridge (2008)
2. Hafiz, A.R., Alnajjar, F., Kazuyuki, M.: A New Dynamic Edge Detection toward Better Human-Robot Interaction. In: Kim, J.-H., et al. (eds.) *FIRA 2009*. LNCS, vol. 5744, pp. 44–52. Springer, Heidelberg (2009)
3. Floreano, D., Zufferey, J.C., Nicoud, J.D.: From wheels to wings with evolutionary spiking neurons. *Artificial Life* 11(12), 121–138 (2005)
4. McClelland, J.: How far can you go with Hebbian learning, and when does it lead you astray. In: Munakata, Y., Johnson, M.H. (eds.) *Attention and Performance XXI: Processes of Change in Brain and Cognitive Development*. Oxford University Press, Oxford (2005)
5. Alnajjar, F., Murase, K.: A simple adaptive controller for autonomous mobile robot: An Aplysia-like spiking neural network with one hidden-layer neuron and spike timing-dependent plasticity. *Adaptive Behavior* 16(5), 306–324 (2008)
6. Bentivegna, D.C., Atkeson, C.G., Ude, A., Cheng, G.: Learning to act from observation and practice. *International Journal of Humanoid Robotics* 1(4), 585–611 (2004)
7. Zin, I., Alnajjar, F., Murase, K.: Adaptation of Real Autonomous Mobile Robot in Complex Environment using Pattern Association Network Controller (PAN-C). *Journal of Advanced Computational Intelligence and Intelligent Informatics* 13(3) (in press 2009)
8. Likhachev, M., Kaess, M., Arkin, R.C.: Learning Behavioral Parameterization Using Spatio-Temporal Case-Based Reasoning. In: *Proceedings IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1282–1289 (2002)
9. Fu, L.M., Hsu, H.H., Principe, J.C.: Incremental backpropagation learning networks. *IEEE Transactions on Neural Networks* 7, 757–761 (1996)
10. Guan, S.U., Li, S.: Incremental Learning with Respect to New Incoming Input Attributes. *Neural Processing Letters* 14(3), 241–260 (2001)
11. Lehtokangas, M.: Modelling with constructive backpropagation. *Neural Networks* 12(4-5), 707–716 (1999)
12. <http://www.atr-robot.com/product/r2/robo-r2.html>

Cellular Neural Networks Template Training System Using Iterative Annealing Optimization Technique on ACE16k Chip

Selcuk Sevgen, Eylem Yucel, and Sabri Arik

Department of Computer Engineering, Istanbul University
34320, Avcilar, Istanbul, Turkey

Abstract. Cellular neural networks proved to be a useful parallel computing system for image processing applications. Cellular neural networks (CNNs) constitute a class of recurrent and locally coupled arrays of identical cells. The connectivity among the cells is determined by a set of parameters called templates. CNN templates are the key parameters to perform a desired task. One of the challenging problems in designing templates is to find the optimal template that functions appropriately for the solution of the intended problem. In this paper, we have implemented the Iterative Annealing Optimization Method on the analog CNN chip to find an optimum template by training a randomly selected initial template. We have been able to show that the proposed system is efficient to find the suitable template for some specific image processing applications.

Keywords: CNNs, Iterative Annealing, ACE16k, Template Training.

1 Introduction

The key feature of a Cellular Neural Network (CNN), introduced in [1], is that it is a locally interconnected analog processor array. Since CNN has two dimensional (2D) grid structure, it is a suitable platform for developing image processing algorithms. Based on the mathematical modeling of CNNs, a programmable CNN, called CNN universal machine (CNN-UM) [3] has been developed. Since these chips have huge computational power and capability of parallel processing, it is possible to perform image processing tasks in a high speed in comparison to conventional architectures.

Program instructions called templates have most important role in the CNN applications. The dynamical behavior of a CNN is completely determined by the templates. The design of suitable templates is one of the fundamental tasks in CNN area.

Kozek T. and et al. used Genetic Algorithm for template learning [6]. Bahram M. and et al. developed a learning algorithm based on Back-Propagation [7]. Loncar A. and et al. developed a simulator system called SCNN which uses wide range of training algorithms [8]. All these methods are simulated to validate

the accuracy of the trained templates using a CNN simulator or calculating the dynamics of the cells.

Parameter variation introduced during fabrication process, noise in the electrical components of the cells, imperfect or noisy loading of the input and initial state and temperature variation can cause erroneous behavior in VLSI (Very Large Scale Integration) implementation of ACE16k. Chip-independent methods or simulation systems of generation of robust templates can not avoid such erroneous behaviors and give accurate results on the chip. In order to overcome this problem, we have designed a template training system on ACE16k chip to obtain more stable templates. Iterative Annealing optimization method, is implemented for the training system. The main advantage of using ACE16k chip is that the processing speed is much higher than the speed of the simulation systems.

Proposed training system can process gray level and black-white input images. By using this system, edge and corner detection templates have been trained. Besides, object counting algorithm based on corner detection has been realized on ACE16k.

2 CNNs and Bi-i Cellular Vision System

2.1 Architecture of CNNs

The basic structure of a CNN of size 4x4 is shown in Fig. 1 where each square represents a cell which is the basic unit. In this CNN architecture, every cell is connected only to its neighbouring cells.

Consider a CNN having $M \times N$ cells arranged in M rows and N columns. The cell on the i th row and j th column is denoted by $C(i,j)$. [1]

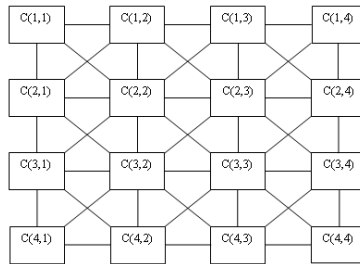


Fig. 1. A two dimensional CNNs of size 4x4

The r -neighbourhood of a cell $C(i,j)$ in a CNN is defined by:

$$N_r(i, j) = \{C(k, l) \mid \max\{|k - i|, |l - j|\} \leq r, 1 \leq k \leq M; 1 \leq l \leq N\} \quad (1)$$

where r is a positive integer.

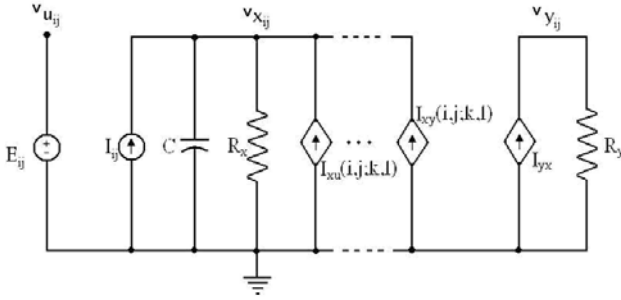


Fig. 2. An example of a cell circuit

A typical example of a cell $C(i,j)$ of a cellular neural network is shown in Fig. 2, where the suffices u,x and y denote the input, state and output, respectively. the node voltage $v_{u_{ij}}$ is called the input of $C(i,j)$ and the node voltage $v_{y_{ij}}$ is called the output of the cell [1].

$$I_{xy}(i,j;k,l) = A(i,j;k,l)v_{y_{kl}}, I_{xu}(i,j;k,l) = B(i,j;k,l)v_{u_{kl}}, \forall C(k,l) \in N_r(i,j) \quad (2)$$

The dynamics of a CNN can be characterised by the following equations [1]:

$$C \frac{dv_{x_{ij}}(t)}{dt} = -\frac{1}{R_x} v_{x_{ij}} + \sum_{C(k,l) \in N_r(i,j)} A(i,j;k,l)v_{y_{kl}}(t) + \sum_{C(k,l) \in N_r(i,j)} B(i,j;k,l)v_{u_{kl}}(t) + I$$

$$\text{where } v_{y_{ij}}(t) = \frac{1}{2}(|v_{x_{ij}}(t)+1| - |v_{x_{ij}}(t)-1|) \quad (3)$$

$$v_{u_{ij}} = E_{ij}, v_{x_{ij}}(0) \leq 1, v_{u_{ij}} \leq 1; 1 \leq k \leq M; 1 \leq l \leq N$$

$$A(i,j;k,l) = A(k,l;i,j), 1 \leq i, k \leq M; 1 \leq j, l \leq N \text{ and } C > 0, R_x > 0 \quad (4)$$

$A(i,j;k,l)$ and $B(i,j;k,l)$ are called the feedback operator and the control operator, respectively.

2.2 Bi-i Cellular Vision System and ACE16k Chip

The Bi-i cellular vision system which contains ACE16k chip and Digital Signal Processor (DSP) is a high-speed, compact and intelligent camera for training. *InstantVision* Libraries and Bi-i SDK (Software Development Kit) are set of C++ programming library for developing Bi-i applications. These libraries can be used with the development environment for the DSP and ACE16k called Code Composer Studio [10].

ACE16k is a CNN-UM implementation. It can be basically described as an array of 128x128 identical, locally interacting, analog processing units designed for high speed image processing tasks. ACE16k is essentially an analog processor (computation is carried out in the analog domain), it can be operated in a fully digital environment [4]. Images can be acquired either by chip specific optical input module or by a digital hosting system [4].

3 Iterative Annealing

Iterative Annealing (IA), a kind of Simulated Annealing [11], is an optimization method specially developed for CNN [13]. The algorithm of Iterative Annealing is shown below:

1. Choose initial values $x_0^k, s_{max}, j_{max}, T_0, \tau, j = 0$
2. Calculate step size $v = (\tau/T_0)^{j_{max}/s_{max}}$
3. $T = T_0, i = 0$
4. $y_i^k = x_i^k + u^k.T$; u^k : Unit distribution U [-0.5,0.5]
5. If $f(\mathcal{V}_i) < f(\mathcal{X}_i)$ then $\mathcal{X}_{i+1} = \mathcal{V}_i$
6. Reduce temperature $T = v.T$
7. $i = i + 1$
8. If $i < (s_{max}/j_{max})$ then Go to 4
9. $j = j + 1$
10. If $(j < j_{max})$ then Go to 3

Where, $f(\vec{x})$: error measure to be minimized, \vec{x} : parameter vector, s_{max} : maximum number of iteration steps, j_{max} : number of reruns to be carried out, T_0 : initial temperature, τ : minimal temperature, v : cooling factor

At every step the temperature is chilling, leading to a decreasing search area until T reaches . Then the process restarts with $T = T_0$. Finally a global minimum is found [13].

4 On Chip Training with Iterative Annealing

Iterative Annealing (IA) method was modified to work on a PC with the ACE16k chip. This means that we can obtain templates which are stable and robust without inaccuracies of CNN-UM hardware realization. ACE16k chip as an external process unit obtains output images for variable template configurations during training process. IA algorithm consists of two loops. The inner loop contains annealing procedure. The outer loop controls iterative behavior. The function to minimize is an error measure calculated between a given reference image and an output image obtained from the chip [14].

Iterative Annealing algorithm can be modified to adapt to ACE16k chip by adding the following steps into the inner loop: 1. Templates are generated by adding $u^k.T$ to the parameter vector to perform them on the chip. 2. Output image is saved for computing error measure. This algorithm generates templates using parameter sets. Then, it loads and runs these templates to ACE16k chip, saves output images and compares them with desired output using error measure function [14].

5 Object Counting Algorithm on Bi-i System

In this section we describe the algorithm for object counting on Bi-i System based on work of Fasih et al [2]. We have developed and adapted the algorithm to ACE16k chip. This algorithm can count objects rapidly in a grey level image (128x128 pixel) because of high-speed offered by Bi-i System. In this algorithm,

we use the north-west corners to count the number of the objects. The algorithm is shown below.

1. Input image is given
2. Threshold operation
3. Opening operation
4. Convex-Hull operation
5. N-W corner detection operation
6. Count number of pixels

6 Experimental Results

We have developed on chip training system by using Iterative Annealing method. We have chosen initial parameters: $T_0 = 10$ and $v = 0.91$ for 100 steps in all optimization procedures. Initial template values are chosen randomly.

In order to test performance of the system, we have tried to learn edge detection template and corner detection template which is not available in TACEIPL. In addition, we have developed an object counting algorithm using the trained N-W corner detection template that we have trained.

6.1 Edge Detection

An edge detection template for the gray-scale images is developed using IAOM. The trained edge detection template is given as follows.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4.54 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} -2.98 & -0.47 & -2.26 \\ 2.07 & 5.66 & 1.74 \\ -3 & 0.74 & -3 \end{bmatrix}, T = -0.96$$

After that we have applied the trained edge detection template to show accuracy of the template on an input image given in Fig. 3a. Result of the edge detection template is shown in Fig. 3b.

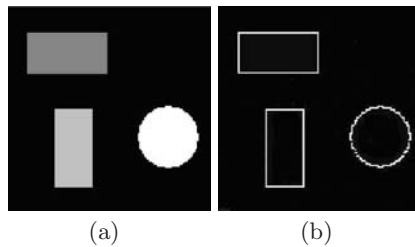


Fig. 3. Edge detection test (a) Sample image (b) Edge detection template result

6.2 Corner Detection

We have also designed corner detection templates. Since a template can not detect all corners, we have tried to train two templates for concave (at least five white neighbor pixels) and convex (at least five black neighbor pixels) corners.

These templates for Concave corner detection template and Convex corner detection template are given in the following, respectively. Here, Fig. 4a, Fig. 4b, Fig. 4c, and Fig. 4d are used as inputs, desired output for concave, desired output for convex images and combined output by Logical OR operation, respectively.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0.97 & -2.3 & 1.37 \\ -1.12 & 5.44 & -0.9 \\ 2.27 & -3 & 1.09 \end{bmatrix}, I = 4.49 \text{ and } A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4.62 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} -1.57 & 0.48 & -2.6 \\ -0.6 & 5.94 & -1.15 \\ -1.25 & -1.53 & 0.07 \end{bmatrix}, I = 4.78$$

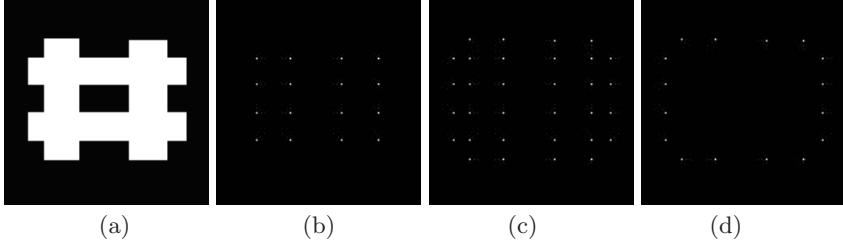


Fig. 4. Corner Detection test (a) Sample image (b) Concave corner detection template result (c) Convex corner detection template result (d) All detected corners

6.3 Comparison of Methods Using Corner Detection

In order to show computational power of ACE16k chip, we have compared our execution times of template operations with a MATLAB implementation of Harris method [5]. Image in Fig. 3a has been used. Our test platform is a PC (Core2Duo 2.0GHz, 2GB RAM). Execution times of template operation and Harris method are 0.000545 s. and 0.0416 s., respectively. Template execution on ACE16k is much faster than Harris method.

6.4 Object Counting

In the first step of the algorithm, input image (Fig. 5a) is converted to binary image by the threshold operator called *ConvLAMtoLLM*. This function in SDK Library, converts a grey image to a binary image on ACE16k [10]. Output image is given in Fig. 5b.

In next step of the algorithm, small objects on binary image are eliminated by *Opening4* function in SDK Library. In addition, this function performs 4-connectivity binary opening using dilation and erosion functions [10]. Noiseless image is given in Fig. 6a.

In the following step, objects on image obtained in previous step are converted to rectangular objects to find North-West corners easily of each object. This operation is performed by *ConvexHull* in SDK Library. Using this function, the objects on the image are involved into a square [10]. Result of this function is shown in Fig.6b.

After ConvexHull operation, North-West corner detection template is applied on image in order to represent each rectangular object as a corner. We trained

this template with the template design tool that we have developed. N-W template and corresponding output image (Fig. 6c) of this template are shown below.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4.76 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 3 & -2.93 & 1.63 \\ -1.87 & 3.51 & -0.19 \\ -0.49 & -0.12 & 2.3 \end{bmatrix}, I = 5.49$$

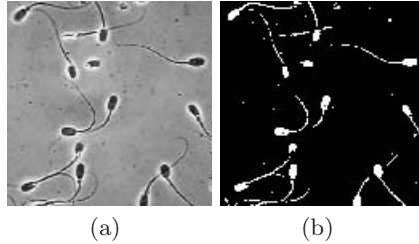


Fig. 5. a) Input Image b) Threshold Result

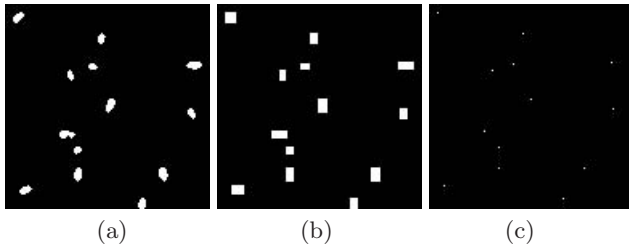


Fig. 6. a) Opening Result b) Convex-Hull Result c) N-W Corner Template Result

Last step of the algorithm is counting white pixels on the image shown in Fig. 6c. This operation is performed using loop operation (for function) in C++ programming language.

We have performed the algorithm on different platforms such as ACE16k and DSP in Bi-i and Matlab in order to show the computational power of ACE16k (Table 1). First three steps of the algorithm are implemented on all platforms. Duration of threshold operation is almost same on these three platforms. The shortest execution times of Opening and Convex-Hull operations belong to ACE16k. Template execution step can not be realized on DSP. In addition, template execution time obtained using MATCNN (A toolbox for CNN implementations on Matlab) [12] was given as reference; because N-W corner template does not work on Matlab. This template was trained to work on ACE16k.

Table 1. Execution times of counting algorithms on different platforms

	ACE16k	DSP	Matlab
Threshold	641 μ s	841 μ s	670 μ s
Opening	425 μ s	1673 μ s	36000 μ s
Convex-Hull	1362 μ s	96154 μ s	9200 μ s
N-W Corner	423 μ s	-	390000 μ s
Counting	4169 μ s	4169 μ s	760 μ s
Total	7020 μ s = 0.00702s	102837 μ s = 0.102s	436630 μ s = 0.436s

Acknowledgments. This work was supported by the Scientific and Technological Research Council of Turkey, under Project 104E024.

References

1. Chua, L.O., Yang, L.: Cellular neural networks: Theory. *IEEE Trans. Circuits Syst.*, 1257–1272 (1998)
2. Fasih, A., Chedjou, J., Kyamakya, K.: Ultra Fast Object Counting Based-on Cellular Neural Network. In: *First International Workshop on Nonlinear Dynamics and Synchronization (INDS 2008)*, pp. 181–183 (2008)
3. Roska, T., Chua, L.O.: The CNN Universal Machine: An Analogic Array Computer. *IEEE Transactions on Circuits and Systems- II: Analog and Digital Signal Processing*, 163–173 (1993)
4. Liñán, G., Domínguez-Castro, R., Espejo, S., Rodríguez-Vázquez, A.: ACE16k: A programmable focal plane vision processor with 128x128 resolution. In: *Eur. Conf. Circuit Theory and Design*, pp. 345–348 (2001)
5. Harris, C., Stephens, M.: A Combined Corner and Edge Detector. In: *Alvey Vision Conference*, pp. 147–151 (1988)
6. Kozek, T., Roska, T., Chua, L.O.: Genetic Algorithm for CNN Template Learning. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 392–402 (1993)
7. Bahram, M., Cheng, Z., Moschytz, G.S.: Learning Algorithms for Cellular Neural Networks. In: *ISCAS 1998*, pp. 159–162 (1988)
8. Loncar, A., Kunz, R., Tetzlaff, R.: SCNN 2000 - Part I: Basic Structure and Features of the Simulation System for Cellular Neural Networks. In: *IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, pp. 123–128 (2000)
9. Chua, L.O., Roska, T.: *Cellular Neural Networks and Visual Computing: Foundation and Applications*. Cambridge University Press, Cambridge (2002)
10. Bi-i Vision System: User Manual
11. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. *Science* 220(4598), 671–679 (1983)
12. <http://www.mathworks.com>
13. Feiden, D., Tetzlaff, R.: Iterative annealing a new efficient optimization method for cellular neural networks. *Image Processing*, 549–552 (2001)
14. Feiden, D., Tetzlaff, R.: On-Chip Training for Cellular Neural Networks using Iterative Annealing. *VLSI circuits and systems* 470–477 (2003)

Estimation of Driving Phase by Modeling Brake Pressure Signals

Hiroki Mima¹, Kazushi Ikeda¹, Tomohiro Shibata¹,
Naoki Fukaya², Kentaro Hitomi², and Takashi Bando²

¹ Nara Institute of Science and Technology, Ikoma, Nara 630-0192 Japan
{hiroki-m,kazushi,tom}@is.naist.jp

² DENSO CORPORATION, Kariya, Aichi 448-8661 Japan
{naoki_fukaya,kentarou_hitomi,takashi_bandou}@denso.co.jp

Abstract. It is important for a driver-assist system to know the phase of the driver, that is, safety or danger. This paper proposes two methods for estimating the driver's phase by applying machine learning techniques to the sequences of brake signals. One method models the signal set with a mixture of Gaussians, where a Gaussian corresponds to a phase. The other method classifies a segment of the brake sequence to one of the hidden Markov models, each of which represents a phase. These methods are validated with experimental data, and are shown to be consistent with each other for the collected data from an unconstrained drive.

1 Introduction

When a driver follows another vehicle, he/she has to maintain a safe following distance to avoid rear-end collisions. Such collisions account for a large portion of injurious/non-injurious accidents, a fact which has motivated the past studies on driver support systems that warn drivers in advance about possible collisions [1,2].

A naive idea for predicting a collision is to evaluate a collision risk and alert the driver when the risk is high. Several risk indices have been proposed from psychological or mechanical viewpoints [3,4,5,6]. These indices take into account the braking response times of drivers or Newtonian mechanics of the two vehicles. However, the risk index has to match the driver's risk perception because the driver uses the brake pedal only when he/she finds it necessary. The risk perception threshold differs from driver to driver, and therefore, the system has to adapt the alertness level to the driver's preference, which must be extracted from the driving data.

The challenging aspect of this problem is to extract this information from the driving data. In order to alert the driver according to past data, we need to detect anomalies in the data. In other words, to ensure that the system only learns "good data", the system should exclude from the training dataset any data collected in dangerous situations.

One method to remove bad data is to estimate the driver's phase, i.e. safety or danger [7,8]. If we know the phase of the driver, then we can take into account the quality of the training data and ignore the data from dangerous situations.

In this paper, we show two methods to estimate the driver's phase using the master cylinder (M/C) pressure of the brake. One advantage of using the brake signal is the ease of onboard measurement. Another advantage is the fact that it represents an intentional operation of the driver to avoid a risk, in contrast to the relative distance between cars or the velocity of the car [9]. The brake signals are given as a sequence consisting of positive sub-sequences which successively take positive values and intervals during which the pressure is zero. We divide the sequence into positive sub-sequences and consider which phase each sub-sequence belongs to by making statistical models of the sub-sequence.

The first method employs Gaussian mixture models (GMMs) [10]. Since the phases of the driver are difficult to define explicitly, we construct a GMM from the dataset in an unsupervised manner and regard one Gaussian as one phase. The second method employs hidden Markov models (HMMs). It is likely that some sub-sequences in a phase are long and others are short. Therefore, an HMM seems suitable for modeling sub-sequences with a variety of lengths.

The rest of the paper is organized as follows. Section 2 describes the details of our collected data. Sections 3 and 4 respectively give brief introductions of GMMs and HMMs, as well as how to model the driving data and the results of experiments. Finally, discussions and conclusions are given in Section 5.

2 Driving Data

We collected the driving data under unconstrained conditions, where we measured the velocity of the vehicle, its distance from the preceding vehicle, and the amount of throttle and M/C pressure of the brake. The driver, who has more than 25 years of driving experience, is the same in all the experiments. He made a round-trip in a suburb, during morning with good weather conditions. We collected 129 sub-sequences in the outgoing portion of the trip and 134 in the incoming portion.

Note that only the brake signal is used in this paper, but the use of others may help to improve our methods in the future.

3 Clustering with a Gaussian Mixture Model

In a GMM, the k -th Gaussian is chosen with probability π_k :

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \Sigma_k), \quad (1)$$

where $\boldsymbol{\mu}_k$ and Σ_k denote the mean and the variance of the k -th Gaussian. Introducing a hidden random variable vector \mathbf{z} , which has the 1-of- K representation with its k -th element having probability π_k , (1) is rewritten as

$$p(\mathbf{x}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) \quad (2)$$

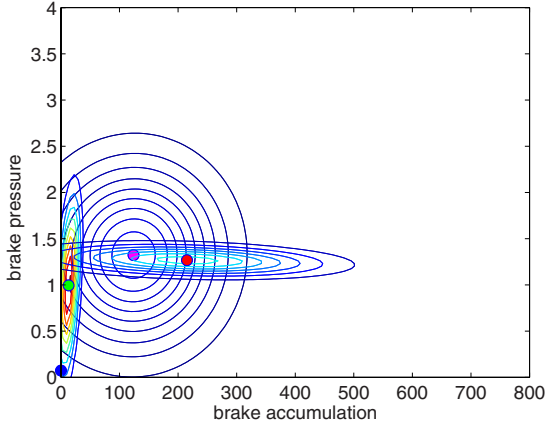


Fig. 1. Iso-probability contour of the densities of the Gaussian components

where

$$p(\mathbf{z}) = \prod_{k=1}^K \pi_k^{z_k}, \quad p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^K \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)^{z_k}. \quad (3)$$

Given a set of data, $\{\mathbf{x}(t)\}_{t=1}^T$, a GMM fits its parameters, π_k , $\boldsymbol{\mu}_k$ and Σ_k to the dataset and estimates the hidden variable vectors, $\{\mathbf{z}(t)\}_{t=1}^T$ using a learning method (e.g. the EM algorithm). Since $\mathbf{z}(t)$ expresses which Gaussian produced the data $\mathbf{x}(t)$, we can classify $\mathbf{x}(t)$ to one of the Gaussian distributions which has the largest $z_k(t)$ in $\{z_k(t)\}_{k=1}^K$.

In our GMM-based analysis, the input vector at time t has two dimensions: the M/C pressure of the brake at time t , and its accumulation in the sub-sequence to which the brake signal belongs. When the pressure is zero, the accumulation is also set to zero. Note that the cardinality of the dataset is 26,751.

Fig. 1 shows the densities of the four Gaussian components trained with the dataset, where four circles show the centers of the components and the ellipses show the contours describing the variances of the Gaussians. We can find the following four categories in the figure:

1. Short-term operations (green).
2. Long-term operations with a large constant pressure (red).
3. Middle-term operations with a variety of pressures (purple).
4. No operations (blue).

Fig. 2 shows the result of the classification, where the color of each point shows the class. We see some curves which correspond to sub-sequences, however, all the data in a sub-sequence do not belong to the same category because each data in a sub-sequence is classified separately. Fig. 3 shows this phenomenon more

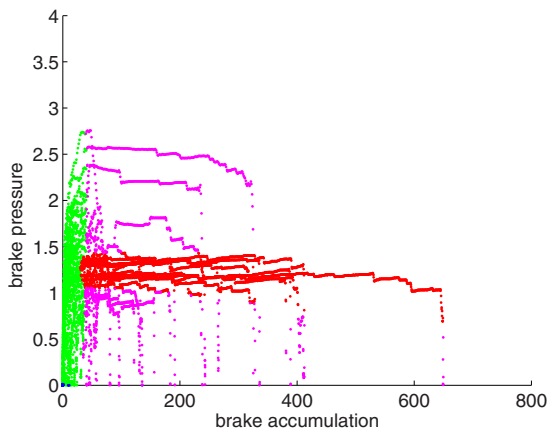


Fig. 2. Clustering of the data with the GMM

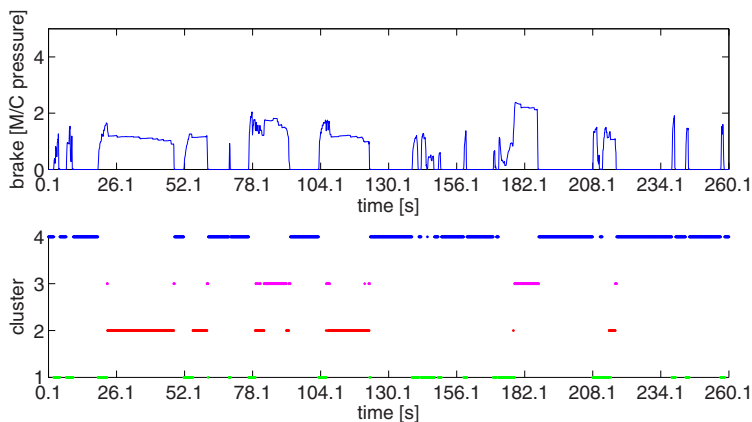


Fig. 3. Clustering of the data with the GMM

clearly, where the upper plot represents a brake signal sequence and the lower plot represents the phase to which the data was classified. In the next section, we propose a method to cope with this problem.

4 Discrimination with a Hidden Markov Model

The problem to be solved in this paper is to estimate the phase to which each sub-sequence belongs. Note that the sub-sequences of brake signals resembles speech signals, in that the duration varies among sub-sequences [11]. Hence, we propose to apply a similar HMM to the sub-sequence classification problem.

We first formulate a general HMM using the hidden variable vector $\mathbf{z}(t)$ which has the 1-of- K representation as a GMM. Fig. 4 is a graphical model of an

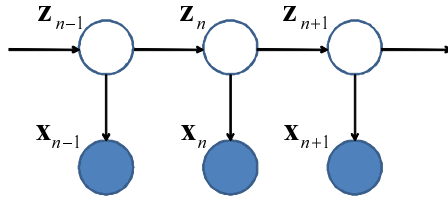


Fig. 4. A graphical model of an HMM

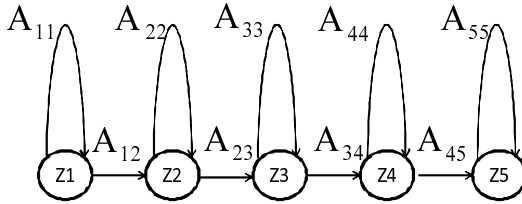


Fig. 5. A state transition graph of a left-to-right HMM

HMM, where the dependencies among the variables are described by arrows. The hidden variable vector $\mathbf{z}(t)$, i.e. the state, depends only on the state $\mathbf{z}(t - 1)$ at the previous time:

$$p(\mathbf{z}(t)|\mathbf{z}(t - 1), A) = \prod_{k=1}^K \prod_{j=1}^K A_{kj}^{z_j^{(t-1)}z_k^{(t)}}, \tag{4}$$

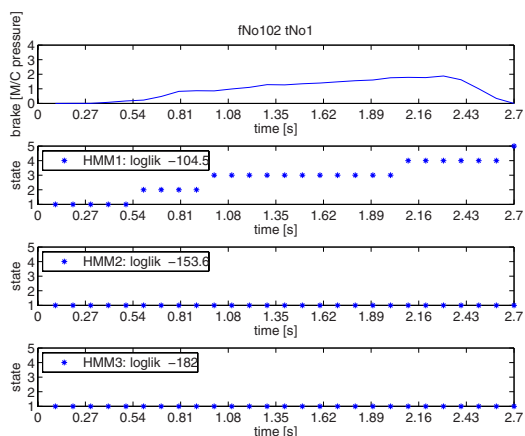
where A_{kj} represents the probability that the state changes from j to k and $A \equiv \{A_{kj}\}$ is the state transition matrix. Our method restricted the class of HMMs to the left-to-right HMMs (Fig. 5), as is done in many applications for sequences. The HMM in our model has five nodes, where the number of nodes was determined by exhaustive experiments. The transition matrix for this class is lower bidiagonal, i.e. the matrix has non-zero entries at the main and lower diagonals.

The output $\mathbf{x}(t)$ depends only on the current state $\mathbf{z}(t)$:

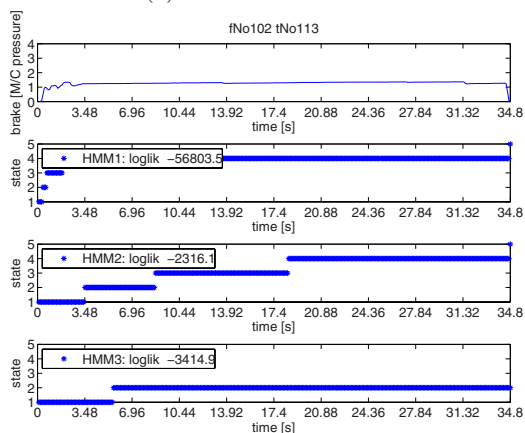
$$p(\mathbf{x}(t)|\mathbf{z}(t)) = \prod_{k=1}^K p(\mathbf{x}(t)|\phi_k)^{z_k^{(t)}}, \tag{5}$$

Table 1. The number of sub-sequences in each class

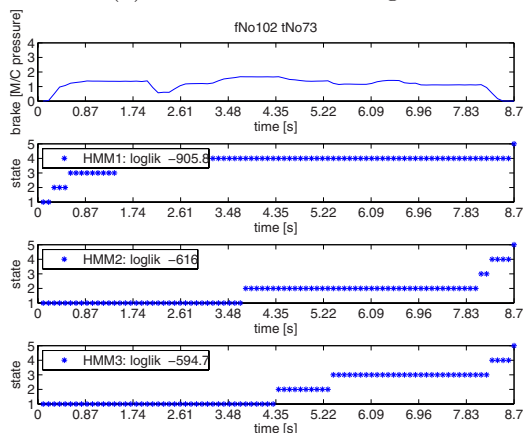
	# of sub-sequences
Phase 1	86
Phase 2	26
Phase 3	16
Phase 4	1



(a) Brake for closeness



(b) Brake for red traffic light



(c) Brake for turning

Fig. 6. Examples of HMM modeling and classification

where ϕ_k denotes the distribution parameter of the data at the k th state. In our method, we employ a Gaussian with mean $\boldsymbol{\mu}_k$ and variance Σ_k .

Note that an HMM can model given sequences but cannot classify them in an unsupervised manner. Hence, we must give a set of data accompanied with the correct labels or classes. In this paper, we used the result of the GMM classifier mentioned in the previous section and determine the label by the majority rule in each sub-sequence. The 129 sub-sequences in the ongoing drive are labeled as Table 1. Note that the training data should be labeled manually in the future.

Each of the four HMMs were trained to model one of the above subsets using HMM Toolbox for Matlab [12]. The estimated transition matrices for Phases 1 to 3 are respectively

$$\begin{pmatrix} 0.560 & 0 & 0 & 0 & 0 \\ 0.440 & 0.643 & 0 & 0 & 0 \\ 0 & 0.357 & 0.878 & 0 & 0 \\ 0 & 0 & 0.122 & 0.854 & 0 \\ 0 & 0 & 0 & 0.146 & 1 \end{pmatrix}, \quad \begin{pmatrix} 0.975 & 0 & 0 & 0 & 0 \\ 0.025 & 0.975 & 0 & 0 & 0 \\ 0 & 0.025 & 0.979 & 0 & 0 \\ 0 & 0 & 0.021 & 0.724 & 0 \\ 0 & 0 & 0 & 0.276 & 1 \end{pmatrix},$$

$$\begin{pmatrix} 0.965 & 0 & 0 & 0 & 0 \\ 0.035 & 0.980 & 0 & 0 & 0 \\ 0 & 0.020 & 0.989 & 0 & 0 \\ 0 & 0 & 0.011 & 0.988 & 0 \\ 0 & 0 & 0 & 0.012 & 1 \end{pmatrix}.$$

Given a new sub-sequence, we calculate the likelihood that each HMM produces the sub-sequence and classify it as the class with the highest likelihood. We show some examples in Fig. 6. The upper plot of (a) represents the brake signal sequence when the car is approaching a preceding car. The other subplots show the state transitions of each HMM that produces the sub-sequence. In this case, the first model has the highest likelihood, and the sub-sequence is classified as Phase 1. Likewise, (b) and (c) are cases when the car is stopping for a red traffic light and when the car is turning right at a crossing, respectively. Phases 2 and 3 have the highest likelihood in cases (b) and (c), respectively.

We input the 134 sub-sequences from the incoming drive into the classifier to evaluate the classification ability of the above model, For 86.4 % of the sub-sequences, the output of the HMM classifier agreed to that of the GMM classifier with the majority rule. Since the datasets are independently collected, this result implies that the method has a high generalization ability.

5 Conclusions

We proposed two methods to estimate the driver's phase from the M/C pressure of the brake, which make it possible for a driver-assist system to qualify the driving data. One method is based on GMMs and classifies the two-dimensional brake signal and its accumulation in an unsupervised manner. However, this method does not take into account the fact that the brake signal is a sequence.

The other method is based on HMMs and assigns a sub-sequence to one of the pre-determined phases. This method is more suitable to analyze time-series but requires labeled training data in advance. These disadvantages will be reevaluated in future studies.

Acknowledgments. This study is supported in part by a Grant-in-Aid for Scientific Research (18300078) from MEXT, Japan.

References

1. Barber, P., Clarke, N.: Advanced collision warning systems. IEE Colloquium 234, 2/1–9 (1998)
2. Piao, J., McDonald, M.: Advanced driver assistance systems from autonomous to cooperative approach. *Transport Review* 28(5), 659–684 (2008)
3. Lee, D.N.: A theory of visual control of braking based on information about time-to-collision. *Perception* 5, 437–459 (1976)
4. Kitajima, S., Marumo, Y., Hiraoka, T., Itoh, M.: Comparison of evaluation indices for estimating driver's risk perception of rear-end collision. *JARI Research Journal* 30(9), 495–498 (2008)
5. Kitajima, S., Kubo, N., Arai, T., Katayama, T.: Reproduction of rear-end collision risk based on data acquired by drive video recorder and verification of driver's brake operation. *JSAE Trans.* 39(6), 205–210 (2008)
6. Mima, H., Ikeda, K., Shibata, T., Fukaya, N., Hitomi, K., Bando, T.: A rear-end collision warning system for drivers with support vector machines. In: *Proc. IEEE Workshop on Statistical Signal Processing* (in press, 2009)
7. Kumagai, T., Akamatsu, M.: Prediction of human driving behavior using dynamic bayesian networks. *IEICE Trans. Information and Systems* E89-D, 857–860 (2006)
8. McCall, J.C., Trivedi, M.M.: Driver behavior and situation aware brake assistance for intelligent vehicles. *Proc. of IEEE* 95(2), 374–387 (2007)
9. Igarashi, K., Miyajima, C., Ito, K., Takeda, K., Itakura, F., Abut, H.: Biometric identification using driving behavioral signals. In: *Proc. IEEE Int'l Conf. on Multimedia and Expo* (2004)
10. Reynolds, D., Rose, R.: Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Trans. Speech and Audio Processing* 3(1), 72–83 (1995)
11. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of IEEE* 77(2), 257–286 (1989)
12. Murphy, K.: Hidden Markov model toolbox for Matlab, <http://www.ai.mit.edu/murphyk/Software/HMM/hmm.html>

Optimal Hyperparameters for Generalized Learning and Knowledge Discovery in Variational Bayes

Daisuke Kaji^{1,2} and Sumio Watanabe³

¹ Computational Intelligence and System Science, Tokyo Institute of Technology, Mailbox R2-5, 4259 Nagatsuda, Midori-ku, Yokohama 226-8503, Japan

² Konicaminolta Medical and Graphic, INC. 2970 Ishikawa-machi, Hachioji-shi, Tokyo 192-8505, Japan

³ Precision and Intelligence Laboratory, Tokyo Institute of Technology, 4259 Nagatsuda, Midori-ku, Yokohama 226-8503, Japan

Abstract. Variational Bayes learning is widely used in statistical models that contain hidden variables, for example, normal mixtures, binomial mixtures, and hidden Markov models. To derive the variational Bayes learning algorithm, we need to determine the hyperparameters in the *a priori* distribution. In the present paper, we propose two different methods by which to optimize the hyperparameters for the two different purposes. In the first method, the hyperparameter is determined for minimization of the generalization error. In the second method, the hyperparameter is chosen so that the unknown hidden structure in the data can be discovered. Experiments are conducted to show that the optimal hyperparameters are different for the generalized learning and knowledge discovery.

1 Introduction

Variational Bayes learning is widely being used in statistical models that contain hidden variables, because its generalization performance is as good as that of Bayes estimation, and its computational costs is as small as those of the EM algorithms. For example, Variational Bayes learning has been applied to information science, pattern recognition, artificial intelligence, and bioinformatics.

In order to derive the variational Bayes learning algorithm, we need to determine the hyperparameters contained in the *a priori* distribution, because the recursive procedure of the variational Bayes explicitly contains the hyperparameters. However, a method by which to control the hyperparameter has not yet been established.

In the present paper, we propose that two different methods of hyperparameter optimization are necessary for the two different purposes. The first method involves minimizing the generalization error so that the probability distribution of the information source is most accurately estimated. The second method involves extracting the hidden minority structure from the given data in order to discover unknown knowledge.

In the first method, we investigate how to control the hyperparameter using the information criteria of the minimum free energy and the minimum generalization error. Although these criteria give different hyperparameters, the hyperparameters are not exceedingly different.

In the second method, we attempt to find as many unknown hidden structures as possible from the given data. The optimal hyperparameters for such a purpose differ from those for the minimum free energy and generalization error. In other words, the hyperparameter that is appropriate for the knowledge discovery is not optimal for the minimum generalization error.

In the experiment, we use a Bernoulli mixture model, and demonstrate that different hyperparameters must be used in variational Bayes learning according to the purposes.

2 Bernoulli Mixture

In the present paper, we investigate variational Bayes learning in the Bernoulli mixture, which is widely used for the analysis of multidimensional binary data. The Bernoulli mixture is known as the latent class analysis [3,10]. The Bernoulli distribution is given by the following conditional probability density function,

$$B(\mathbf{x}|\boldsymbol{\mu}) = \prod_{i=1}^M \mu_i^{x_i} (1 - \mu_i)^{(1-x_i)},$$

where $\mathbf{x} = (x_1, \dots, x_M)^T$ is a datum, $\boldsymbol{\mu} = (\mu_1, \dots, \mu_M)^T$ is a parameter and M is the dimension of the datum. Then the Bernoulli mixture is defined by

$$p(\mathbf{x}|\boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{k=1}^K \pi_k B(\mathbf{x}|\boldsymbol{\theta}_k), \quad (1)$$

where $\boldsymbol{\pi}$ denotes the mixture ratio of $B(\mathbf{x}|\boldsymbol{\theta}_k)$ and K is the number of mixtures, $\boldsymbol{\theta}$ is $K \times M$ parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K)$. Here, we introduce the hidden parameters associated with the datum \mathbf{x} . The hidden parameter \mathbf{z} denotes the distribution that generates the datum \mathbf{x} , and \mathbf{z} is expressed as a competitive vector $\mathbf{z} = (0, \dots, 1, \dots, 0)$.

Here, we introduce the Dirichlet and Beta distribution as the conjugate prior distributions of the hidden parameter \mathbf{z} and datum \mathbf{x} , respectively. Then the distributions of $\mathbf{Z} = (z_1, \dots, z_N)$, $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$, $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ are given, respectively, by

$$p(\mathbf{Z}|\boldsymbol{\pi}) = \prod_{n=1}^N \prod_{k=1}^K \pi_k^{z_{nk}}, \quad (2)$$

$$p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\theta}) = \prod_{n=1}^N \prod_{k=1}^K \left(\prod_{m=1}^M \theta_{km}^{x_{nm}} (1 - \theta_{km})^{(1-x_{nm})} \right)^{z_{nk}}, \quad (3)$$

$$p(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi}|a) = \frac{\Gamma(Ka)}{\Gamma(a)^K} \prod_k \pi_k^{a-1}, \tag{4}$$

$$p(\boldsymbol{\theta}) = \prod_{k=1}^K \prod_{m=1}^M \text{Beta}(\theta_{km}|b) = \prod_{k=1}^K \prod_{m=1}^M \left(\frac{\Gamma(2b)}{\Gamma(b)^2} \theta_{km}^{b-1} (1 - \theta_{km})^{b-1} \right), \tag{5}$$

where (a, b) is the set of parameter in the *a priori* distributions $p(\boldsymbol{\pi})$ and $p(\boldsymbol{\theta})$ respectively and these parameters are called hyperparameters. In the present paper, we investigate two different methods by which to determine the hyperparameters.

3 Variational Bayes Algorithm

3.1 General Framework of the Variational Bayes Algorithm

In this section, \mathbf{Y} denotes all hidden variables, including parameters, and \mathbf{X} denotes all variables that are observable. The following equation relates an arbitrary probability distribution $q(\mathbf{Y})$ and the *a posteriori* distribution $p(\mathbf{Y}|\mathbf{X})$:

$$F(\mathbf{X}) = \bar{F}[q(\mathbf{Y})] + KL(q(\mathbf{Y})\|p(\mathbf{Y}|\mathbf{X})), \tag{6}$$

where the free energy F , the variational free energy \bar{F} and the Kullback-Leibler divergence KL are given as follows:

$$F(\mathbf{X}) = -\log \int p(\mathbf{X}, \mathbf{Y}) d\mathbf{Y} = -\log p(\mathbf{X}),$$

$$\bar{F}[q(\mathbf{Y})] = \int q(\mathbf{Y}) \log \frac{q(\mathbf{Y})}{p(\mathbf{X}, \mathbf{Y})} d\mathbf{Y},$$

$$KL(q(\mathbf{Y})\|p(\mathbf{Y}|\mathbf{X})) = \int q(\mathbf{Y}) \log \frac{q(\mathbf{Y})}{p(\mathbf{Y}|\mathbf{X})} d\mathbf{Y}.$$

The variational posterior distribution $q(\mathbf{Y})$ is optimized by minimization of $\bar{F}[q(\mathbf{Y})]$, which is equivalent to the minimization of the Kullback-Leibler divergence between $q(\mathbf{Y})$ and the true posterior $p(\mathbf{Y}|\mathbf{X})$. Here the variational Bayesian approach assumes that the parameters and hidden variables are conditionally independent of each other in order to obtain a computationally tractable posterior. Hence, when we denote \mathbf{w} as parameters, then $q(\mathbf{Y})$ is expressed as

$$q(\mathbf{Y}) = q(\mathbf{Z}, \mathbf{w}) = q_1(\mathbf{Z})q_2(\mathbf{w}).$$

Minimization of the functional $\bar{F}[q(\mathbf{Y})]$ with respect to the above q_1 and q_2 can be performed by using variational methods. By solving the minimization problem under the constraints $\int_{\mathbf{Z}} q_1(\mathbf{Z}) = 1, \int q_2(\mathbf{w}) d\mathbf{w} = 1$, we can obtain the following equations,

$$\log q_1(\mathbf{Z}) = E_{q_2}[\log P(\mathbf{X}, \mathbf{Z}, \mathbf{w})] + C_1, \tag{7}$$

$$\log q_2(\mathbf{w}) = E_{q_1}[\log P(\mathbf{X}, \mathbf{Z}, \mathbf{w})] + C_2, \tag{8}$$

where C_1, C_2 are the normalization constants.

3.2 Variational Bayes Algorithm for Bernoulli Mixture

The variational Bayes learning is carried out by recursive calculation of (7) and (8). By calculating (7) and (8) for Bernoulli mixture under the setting we described in the section 2 we obtain the following algorithm,

VB e-step

$$\begin{aligned} \log \rho_{nk} &= \psi(\alpha_k) - \psi\left(\sum_k^K \alpha_k\right) \\ &+ \sum_{m=1}^M (x_{nm}\psi(\eta_{km}) - x_{nm}\psi(\eta'_{km}) + \psi(\eta'_{km}) - \psi(\eta_{km} + \eta'_{km})) \\ r_{nk} &= \frac{\rho_{nk}}{\sum_{k=1}^K \rho_{nk}} \end{aligned}$$

VB m-step

$$\begin{aligned} N_k &= \sum_{n=1}^N r_{nk}, \quad a_k = a + N_k \\ \eta_{km} &= b + \sum_{n=1}^N r_{nk}x_{nm}, \quad \eta'_{km} = b + \sum_{n=1}^N r_{nk}(1 - x_{nm}) \end{aligned}$$

The above algorithm illustrate the update formulae with respect to the hyperparameters of posterior $q_1(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi}|\mathbf{a})$ and $q_2(\boldsymbol{\theta}) = \prod_{k=1}^K \prod_{m=1}^M \text{Beta}(\theta_{km}|\eta_{km}, \eta'_{km})$, and ψ denotes the digamma distribution $\psi(a) \equiv \frac{d}{da} \Gamma(a) = \frac{\Gamma'(a)}{\Gamma(a)}$.

4 Hyperparameter Optimization

4.1 Hyperparameter for Generalized Learning

The free energy F that is minimized by the optimal $q(\mathbf{Y})$ is referred to as the variational free energy. The variational predictive distribution is defined by

$$p(\mathbf{x}|\mathbf{X}) = \int p(\mathbf{x}|\mathbf{w})q_2(\mathbf{w})d\mathbf{w}.$$

The variational generalization error G is defined by the Kullback-Leibler distance between the true distribution $R(\mathbf{x})$ and the variational predictive distribution,

$$G = E \left[\int R(\mathbf{x}) \log \frac{R(\mathbf{x})}{p(\mathbf{x}|\mathbf{X})} d\mathbf{x} \right],$$

where $E[\cdot]$ denotes the expectation value over all sets of training data \mathbf{X} .

Note that mixture models are not regular but singular models from a statistical points of view. Although learning properties of singular models remain

unknown, it has been reported that the asymptotic behaviors of the Bayes free energy and Bayes generalization error are determined by algebraic geometrical structures [4,5,6].

However, since the variational Bayes learning differs from Bayes learning, there remains no mathematical foundation for variational Bayes learning. It has been reported that the variational free energy changes its behavior at $a = \frac{M+1}{2}$ [1]. Actually the learning result has a phase transition at $a = \frac{M+1}{2}$. More specifically, a mixture ratio of redundant components approaches zero in $a < \frac{M+1}{2}$ and the algorithm attempts to express the predictive distribution by using all redundant components evenly in $a > \frac{M+1}{2}$. However the influence of this behavior on the generalization error has not yet been clarified. The variational generalization error has been reported not to have a direct mathematical relationship with the variational free energy [2].

In the following section, we experimentally investigate the generalization problems with respect to the hyperparameters.

4.2 Hyperparameter for Knowledge Discovery

A mixture model such as a Bernoulli mixture is used for the unsupervised clustering in application. In such a cases, knowledge discovery or data mining is emphasized rather than the generalization error. With respect to the parameter setting, the hyperparameters are generally set based on the prior information, if we have any knowledge about the analysis object. In contrast, if we do not use the prior information, a uniform distribution, such as $a = 1, b = 1$ in Bernoulli mixtures, is often adopted. However, we usually have a reason for performing clustering or analysis, even when we do not have any prior information. In other words, we usually have a requirement with respect to the size of the cluster. For example, when we classify the data, we sometimes consider not only main clusters but also small and minority clusters. This case corresponds to the extraction of a minority cluster from, for example, a questionnaire. In a Bernoulli mixture, it would appear that assigning a small b enables minority cluster extraction. In this case, the prior distribution generates 0 or 1 with high probability. As a this result, the predictive distribution becomes adapted to a small cluster that generates a number of specific terms. The above theorem suggests that the combination of parameters $a < \frac{M+1}{2}$ and small b enables both a small number of clusters and the minority cluster to be extracted.

5 Experiments

5.1 Variational Free Energy and Generalization Error

We first investigated the behaviors of the variational free energy and generalization error. We used 1,000 samples in one trial and calculated the experimental expectation values over 100 trials. The free energy is shown in Fig 1, where (a, b) is the set of hyperparameters of the mixture ratio and the Bernoulli distribution. The true distribution was designed to have the parameter described in

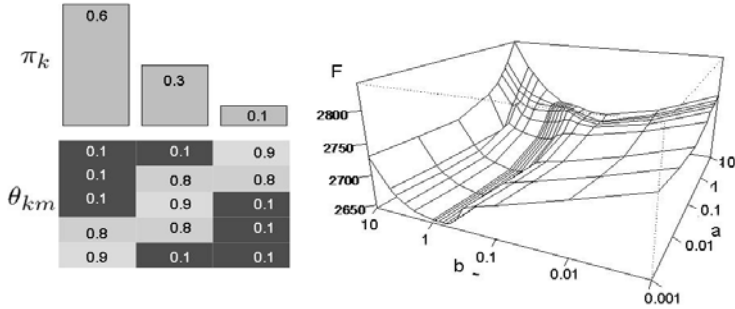


Fig. 1. (left) True distribution and (right) variational free energy (a, b : log scale)

the left-hand of Fig. 1 (the true distribution consists of 3 mixture components and each component is a 7 dimensional Bernoulli distribution, here white indicates the high probability), and the stop condition of the learning is given by “maximum variation of all parameters $< 10^{-3}$ ”. The number of mixture components of learner was set to $K = 10$.

The generalization error appears in Fig. 2. In this case, the minimum point of the variational generalization error corresponds approximately to the minimum point of the variational free energy.

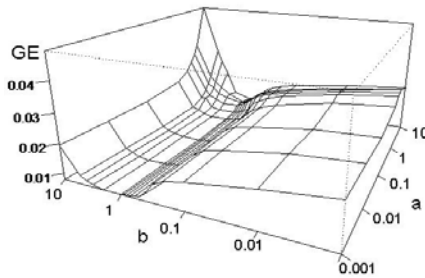


Fig. 2. Generalization error (a, b : log scale)

Peaks in the variational free energy appeared around $(a, b) = (3, 1)$. This phenomenon is thought to be related to the phase transition. In addition, Fig. 2 shows a region of small a and the region around $b = 1$ was stable with respect to the variational generalization error. Therefore, in order to make the generalization error small, the hyperparameter $(a, b) = (\text{small}, 1)$ is recommended.

5.2 Knowledge Discovery

In this section, we investigate a method by which to find the minority cluster. We used the true distribution composed of three mixture components, in which

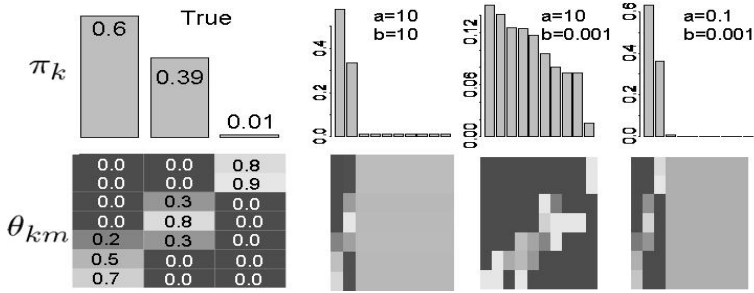


Fig. 3. Extraction of a minority cluster

one component, as a minority cluster, has a mixture ratio of 0.01. The aspect of predictive distribution by average parameters is shown in Fig.3. For example, when both a and b were set large, the learner could not find small clusters. On the other hand, the pair of large a and small b extracted several clusters. When small a and b (right) are chosen, the learner simultaneously reduces the number of clusters and extracts a minority cluster. Consequently, in order to find the minority cluster, the hyperparameter set $(a, b) = (\text{small}, \text{small})$ was appropriate. Although such a set of hyperparameters may not be optimal for the minimum generalization error, because these hyperparameters make too much fitting to the data. However, it is useful for knowledge discovery.

Finally we applied the above setting to the practical data obtained from the following web site “<http://kiwitobes.com/clusters/zebo.txt>” [11]. These data were obtained from a matrix composed of 83 items and 1750 users, in which elements are assigned a value of 1 if some users want (or own or ‘love’) the item, and other elements are assigned a value of 0. The result applied to the data is illustrated in Fig.4 (we listed the items in decreasing order of the probability for each category) where category3 and category4 are large clusters, and category11 is a very small cluster. In this case, the probability of category 11 was expressed as either high

Category1		Category2		Category11	
house	0.46400	laptop	0.19441	xbox 360	0.99997
money	0.19700	house	0.17127	ps3	0.99997
job	0.05883	ipod	0.15085	psp	0.99997
business	0.05241	money	0.11379	ipod	0.33364
clothes	0.05219	computer	0.08726	mansion	0.00003
shoes	0.05175	cell phone	0.07031	sports car	0.00003
friends	0.04112	bike	0.04819	bike	0.00003
big house	0.04047	friends	0.04379	clothes	0.00003
mansion	0.03863	ps3	0.04267	kids	0.00003

Fig. 4. Clustering result for practical data(left column:item, right column:probability)

probability or low probability, i.e., 0.999947 or $3.34E - 05$. This result suggests that category 11 contains a minority cluster that has very similar interests.

6 Conclusion

In the present paper, we proposed two methods for hyperparameter optimization: a method to minimize the generalization error and a method for knowledge discovery. The hyperparameters for the minimum generalization error and the hyperparameters for the minimum variational free energy are not so different. This result suggests the adequacy of selecting the hyperparameters for the minimum generalization error by the variational free energy. On the other hand, small a and b enable us to find minority from the data. We guess that it is possible to obtain the same effect in case of the Gaussian mixture by setting hyperparameters giving small variances to the Gaussian distributions. However, the above hyperparameters are not for minimizing the generalization error. Our experimental results demonstrate that the optimal hyperparameters for the different purposes are different from each other.

Acknowledgment

This work was supported by the Ministry of Education, Science, Sports, and Culture in Japan, Grand-in-aid for scientific research 18079007.

References

1. Watanabe, K., Watanabe, S.: Stochastic complexities of general mixture models in Variational Bayesian Approximation. *Neural Computation* 18(5), 1007–1065 (2006)
2. Nakajima, S., Watanabe, S.: Variational Bayes Solution of Linear Neural Networks and its Generalization Performance. *Neural Computation* 19(4), 1112–1153 (2007)
3. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer, Heidelberg (2006)
4. Watanabe, S.: Algebraic analysis for singular statistical estimation. In: Watanabe, O., Yokomori, T. (eds.) ALT 1999. LNCS (LNAI), vol. 1720, pp. 39–50. Springer, Heidelberg (1999)
5. Watanabe, S.: Algebraic Analysis for Nonidentifiable Learning Machines. *Neural Computation* 13(4), 899–933 (2001)
6. Watanabe, S.: Learning efficiency of redundant neural networks in Bayesian estimation. *IEEE Transactions on Neural Networks* 12(6), 1475–1486 (2001)
7. Hofmann, T.: Probabilistic Latent Semantic Indexing. In: Proc. of SIGIR 1999, pp. 50–57 (1999)
8. Attias, H.: Inferring parameters and structure of latent variable models by variational Bayes. In: *Proceedings of Uncertainty in Artificial Intelligence (UAI 1999)* (1999)
9. Beal, M.J.: Variational Algorithms for approximate Bayesian inference. PhD thesis, University College London (2003)
10. Lazarsfeld, P.F., Henry, N.W.: *Latent structure analysis*. Houghton Mifflin, Boston (1968)
11. Segaran, T.: *Programming Collective Intelligence: Building Smart Web 2.0 Applications*. O'Reilly Media, Inc. (2007)

Backpropagation Learning Algorithm for Multilayer Phasor Neural Networks

Gouhei Tanaka* and Kazuyuki Aihara

Institute of Industrial Science, University of Tokyo,
Tokyo 153-8505, Japan

{gouhei, aihara}@sat.t.u-tokyo.ac.jp

<http://www.sat.t.u-tokyo.ac.jp>

Abstract. We present a backpropagation learning algorithm for multilayer feedforward phasor neural networks using a gradient descent method. The state of a phasor neuron takes a complex-valued state on the unit circle in the complex domain. Namely, the state can be identified only by its phase component because the amplitude component is fixed. Due to the circularity of the phase variable, phasor neural networks are useful to deal with periodic and multivalued variables. Under the assumption that the weight coefficients are complex numbers and the activation function is a continuous and differentiable function of a phase variable, we derive an iterative learning algorithm to minimize the output error. In each step of the algorithm, the weight coefficients are updated in the gradient descent direction of the error function landscape. The proposed algorithm is numerically tested in function approximation task. The numerical results suggest that the proposed method has a better generalization ability compared with the other backpropagation algorithm based on linear correction rule.

Keywords: Phasor neural networks, Complex-valued neuron, Learning, Backpropagation, Gradient descent.

1 Introduction

Complex numbers have been widely used for information representation in engineering and physics. Indeed, complex number calculus is convenient to deal with a variety of information, including not only complex-valued data but also multivalued data. For instance, complex-valued representation is suited for waves with amplitude and phase components, while multivalued representation is useful for digital images with multiple colors. Therefore, complex-valued information processing is becoming more and more important with expectation for its wide applications [1]. In such backgrounds, it is understandable that complex-valued neural networks have been widely studied in recent years [2,3].

In the 1990s, learning algorithms for layered complex-valued neural networks were intensively studied. Since a complex function that is bounded and regular

* Corresponding author.

(differentiable) is restricted to a constant function due to Liouville’s theorem, the main concern was to seek an appropriate complex activation function which is effective for learning complex-valued information. A simple generalization of the sigmoid function $g(x)$, typically used in real-valued neural networks, is the real-imaginary type activation function, i.e., $f(z) = g(\text{Re}(z)) + g(\text{Im}(z))$ where z is a complex number. The backpropagation algorithm for complex-valued networks with this activation function can be simply obtained by extending the real-valued case [4]. The complex activation function, in which only the amplitude component of the input is nonlinearly transformed and the phase component is unchanged, is called the amplitude-phase type activation function, i.e., $f(z) = g(|z|) \exp(i\arg(z))$. The backpropagation learning algorithm with this activation function was also derived [5,6].

We focus on a special class of complex-valued neural networks, where the neuronal states are located on the unit circle in the complex domain. It is called a phasor neural network [7,8]. Recurrent networks of phasor neurons have been applied to multistate associative memories [9,10,11]. In phasor neural networks, only the phase component is the information carrier while the amplitude of each neuronal state is fixed at one. By dividing the unit circle into K arcs with equal size, the K boundary points can be used to represent K -valued states. This discrete phasor neuron is also called a multivalued neuron [12].

Recently, a complex activation function with continuous nonlinearity for a phasor neuron has been proposed to improve the capability of the conventional complex-valued Hopfield network based on discrete activation functions [13]. Using the differentiable activation function, we derive a backpropagation learning algorithm based on a gradient descent method for multilayer feedforward phasor neural networks. The presented method is tested in numerical experiments on function approximation task. The numerical results show that the proposed method yields less test errors than the heuristic learning algorithm based on a linear correction rule [14].

2 Activation Function of Phasor Neurons

The complex-signum function [9], which is used for a discrete phasor neuron [7,8] and a multivalued neuron [12], is described as follows:

$$\begin{aligned}
 f_d(z) &= \exp\left(i\frac{2\pi}{K} \left\{ \left[\frac{K\arg(z)}{2\pi} \right] + \frac{1}{2} \right\}\right) \\
 &= \begin{cases} e^{i\pi/K} & (0 \leq \arg(z) < 2\pi/K), \\ \vdots & \vdots \\ e^{i(2\pi k+\pi)/K} & (2\pi k/K \leq \arg(z) < 2\pi(k+1)/K), \\ \vdots & \vdots \\ e^{i(2\pi(K-1)+\pi)/K} & (2\pi(K-1)/K \leq \arg(z) < 2\pi), \end{cases} \quad (1)
 \end{aligned}$$

where $[\cdot]$ indicates the floor function. Equation (1) is called a K -state phasor or a K -valued neuron for an integer K , because the number of discrete neuronal

states is given by K . The complex-signum function is based on a multilevel staircase-like function, which is regarded as a generalization of the two-state step function.

By replacing the discrete multilevel function with its continuous version defined on a circle, the complex-sigmoid function [13] is obtained as follows:

$$f_c(z) = \exp \left\{ i \frac{2\pi}{K} m_K \left(\frac{K \arg(z)}{2\pi} \right) \right\}. \tag{2}$$

The continuous multilevel function with circularity, $m_K : [0, K) \rightarrow [0, K)$, is defined by using the multilevel sigmoid function [15,16] as follows:

$$m_K(x) = \left(\sum_{k=0}^{K-1} g(x - k) \right) - \frac{1}{2} \pmod{K}, \tag{3}$$

where $g(x) = 1/(1 + \exp(-x/\epsilon))$. For any real number u , $u \pmod{K} \equiv u + jK \in [0, K)$ where j is an appropriate integer. It should be noted that the continuous multilevel function is differentiable because $g'(x) = g(x)(1 - g(x))/\epsilon$ but the discrete multilevel function is not. The complex-sigmoid function [2] is a generalization of the complex-signum function [1], i.e., $f_c(z) \rightarrow f_d(z)$ for any z in the limit of $\epsilon \rightarrow 0$.

The complex-valued mapping [2] can be rewritten as follows:

$$\arg(f_c(z)) = \frac{2\pi}{K} m_K \left(\frac{K \arg(z)}{2\pi} \right). \tag{4}$$

Therefore, Eq. [2] is essentially reduced to the following circle map:

$$f_p(\varphi) = \frac{2\pi}{K} \left\{ \sum_{k=0}^{K-1} g \left(\frac{K\varphi}{2\pi} - k \right) - \frac{1}{2} \right\}, \tag{5}$$

where $\varphi \equiv \arg(z) \in [0, 2\pi)$ and $f_p(\varphi) \equiv \arg(f_c(z))$. This reduction is possible because the neuronal state can be identified only by the real-valued phase component φ . In a gradient descent learning method, the activation function requires to be continuous and differentiable. It is more convenient to consider the differential of f_p with respect to φ than that of f_c with respect to z . The differential of the continuous activation function [5] is calculated as:

$$f'_p(\varphi) = \frac{1}{\epsilon} \sum_{k=0}^{K-1} g \left(\frac{K\varphi}{2\pi} - k \right) \left\{ 1 - g \left(\frac{K\varphi}{2\pi} - k \right) \right\}. \tag{6}$$

In the limit of $K \rightarrow \infty$, it is obvious from Eqs. [1] and [2] that $f_d(z) \rightarrow e^{i\arg(z)}$ and $f_c(z) \rightarrow e^{i\arg(z)}$. Similarly, as K goes to the infinity, we obtain the following activation function:

$$f_p(\varphi) = \varphi, \tag{7}$$

$$f'_p(\varphi) = 1, \tag{8}$$

for any $\varphi \in [0, 2\pi)$. Later, we adopt this limit activation function for numerical simulations instead of Eq. [5] because it is better in terms of computation time.

3 Multilayer Feedforward Phasor Neural Networks

For simplicity, we consider three-layer feedforward phasor neural networks including one input, one hidden, and one output layers as schematically illustrated in Fig. 1. The numbers of input, hidden, and output units are denoted by N_j , N_k , and N_l , respectively. The neuronal states of the input, hidden, output layers are represented by $z_j = e^{i\theta_j}$ ($1 \leq j \leq N_j$), $z_k = e^{i\theta_k}$ ($1 \leq k \leq N_k$), and $z_l = e^{i\theta_l}$ ($1 \leq l \leq N_l$), respectively. The weight coefficient of the k th hidden unit for the j th input unit is denoted by w_{kj} , and that of the l th output unit for the k th hidden unit is denoted by w_{lk} .

The weighted sum of inputs to the k th hidden unit is

$$u_k e^{i\varphi_k} \equiv \sum_{j=0}^{N_j} w_{kj} z_j = \sum_{j=0}^{N_j} w_{kj} e^{i\theta_j}, \tag{9}$$

where a dummy unit with $z_0 = 1$ and $\theta_0 = 0$ is introduced for the bias parameters w_{k0} . The output of the k th hidden unit is given as

$$z_k = e^{i\theta_k} = f_c(u_k e^{i\varphi_k}), \tag{10}$$

$$\theta_k = f_p(\varphi_k). \tag{11}$$

Similarly, the weighted sum of inputs to the l th output unit is

$$u_l e^{i\varphi_l} \equiv \sum_{k=0}^{N_k} w_{lk} z_k = \sum_{k=0}^{N_k} w_{lk} e^{i\theta_k}, \tag{12}$$

where the same dummy unit is used again for the bias parameters w_{l0} . Then, the output of the l th output unit is given as

$$z_l = e^{i\theta_l} = f_c(u_l e^{i\varphi_l}), \tag{13}$$

$$\theta_l = f_p(\varphi_l). \tag{14}$$

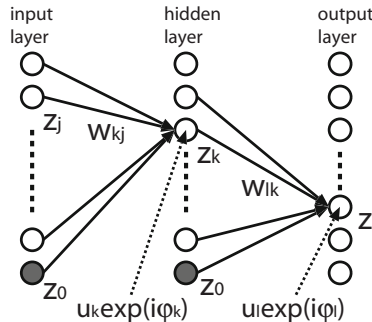


Fig. 1. Structure of a three-layer feedforward phasor neural network consisting of N_j input, N_k hidden, and N_l output units. The filled circles indicate the dummy units for the bias parameters.

All the states of the units can be identified only by the phase components. However, it should be remembered that complex number representation and complex number calculus are still essential for the forward propagation of the multilayer phasor neural network.

4 Learning Algorithm

4.1 Gradient Descent Method

Before learning, the current output of the network with randomly distributed weights is different from the desired output in general. The purpose of learning is to minimize the output error, or the difference between the current and desired outputs. We adopt the method of updating the weight coefficients iteratively.

Suppose that the weight coefficient between the l th output unit and the k th hidden unit is updated as follows:

$$\tilde{w}_{lk} = w_{lk} + \Delta w_{lk}, \quad (15)$$

where $w_{lk} = w_{lk}^R + iw_{lk}^I$ and $\tilde{w}_{lk} = \tilde{w}_{lk}^R + i\tilde{w}_{lk}^I$ are the current and updated weights, respectively. The real and imaginary parts of a complex number are indicated by the superscripts R and I , respectively. In a gradient descent method, the variations of the weights are proportional to the negative value of the gradient descent as follows:

$$\Delta w_{lk}^R = -\eta \frac{\partial E}{\partial w_{lk}^R}, \quad (16)$$

$$\Delta w_{lk}^I = -\eta \frac{\partial E}{\partial w_{lk}^I}, \quad (17)$$

where η is the learning rate.

We assume that training data are given as a combination of N_j -dimensional input vector and N_l -dimensional output vector $\hat{\mathbf{z}} = (\hat{z}_1, \dots, \hat{z}_{N_l})$ where $\hat{z}_l = e^{i\hat{\theta}_l}$. The current network output for the input training vector is denoted by $\mathbf{z} = (z_1, \dots, z_{N_l})$ where $z_l = e^{i\theta_l}$. The aim of learning is to minimize the output error:

$$\begin{aligned} E &= \frac{1}{2} \|\mathbf{z} - \hat{\mathbf{z}}\|^2 \\ &= 1 - \sum_{l=1}^{N_l} \cos(\theta_l - \hat{\theta}_l). \end{aligned} \quad (18)$$

Using the chain rule, we obtain

$$\frac{\partial E}{\partial w_{lk}^R} = \frac{\partial E}{\partial \theta_l} \frac{d\theta_l}{d\varphi_l} \frac{\partial \varphi_l}{\partial w_{lk}^R}, \quad (19)$$

$$\frac{\partial E}{\partial w_{lk}^I} = \frac{\partial E}{\partial \theta_l} \frac{d\theta_l}{d\varphi_l} \frac{\partial \varphi_l}{\partial w_{lk}^I}, \quad (20)$$

where

$$\frac{\partial E}{\partial \theta_l} = \sin(\theta_l - \hat{\theta}_l), \quad (21)$$

$$\frac{d\theta_l}{d\varphi_l} = f'_p(\varphi_l). \quad (22)$$

In order to calculate the remaining terms $\partial\varphi_l/\partial w_{lk}^R$ in Eq. (19) and $\partial\varphi_l/\partial w_{lk}^I$ in Eq. (20), we suppose that the weights w_{lk} ($k = 1, \dots, N_k$) and the bias w_{l0} of the l th output unit are updated as follows:

$$w_{lk} \rightarrow w_{lk} + (\Delta w_{lk}^A + i\Delta w_{lk}^P)e^{i(\varphi_l - \theta_k)}. \quad (23)$$

This update results in the variation of the weighted sum of inputs as follows:

$$\Delta(u_l e^{i\varphi_l}) = (\Delta w_{lk})z_k = (\Delta w_{lk}^A + i\Delta w_{lk}^P)e^{i\varphi_l}. \quad (24)$$

The terms Δw_{lk}^A and Δw_{lk}^P are independently related to the variation of the weighted sum of inputs in the directions of amplitude and phase, respectively [6]. Therefore, it follows that

$$\frac{du_l}{dw_{lk}^A} = 1, \quad (25)$$

$$\frac{d\varphi_l}{dw_{lk}^P} = \frac{1}{u_l}. \quad (26)$$

We can rewrite Eq. (23) as follows:

$$\Delta w_{lk}^R + i\Delta w_{lk}^I = (\Delta w_{lk}^A + i\Delta w_{lk}^P)e^{i(\varphi_l - \theta_k)}. \quad (27)$$

Comparing the real and imaginary parts in both sides separately, we obtain the following relation:

$$\Delta w_{lk}^R = \cos(\varphi_l - \theta_k)\Delta w_{lk}^A - \sin(\varphi_l - \theta_k)\Delta w_{lk}^P, \quad (28)$$

$$\Delta w_{lk}^I = \sin(\varphi_l - \theta_k)\Delta w_{lk}^A + \cos(\varphi_l - \theta_k)\Delta w_{lk}^P. \quad (29)$$

Since the vector $(\Delta w_{lk}^R, \Delta w_{lk}^I)^T$ is a rotation of the vector $(\Delta w_{lk}^A, \Delta w_{lk}^P)^T$ with angle $\varphi_l - \theta_k$, the inverse rotation yields

$$\Delta w_{lk}^A = \cos(\varphi_l - \theta_k)\Delta w_{lk}^R + \sin(\varphi_l - \theta_k)\Delta w_{lk}^I, \quad (30)$$

$$\Delta w_{lk}^P = -\sin(\varphi_l - \theta_k)\Delta w_{lk}^R + \cos(\varphi_l - \theta_k)\Delta w_{lk}^I. \quad (31)$$

These equations lead to

$$\frac{\partial w_{lk}^P}{\partial w_{lk}^R} = -\sin(\varphi_l - \theta_k), \quad (32)$$

$$\frac{\partial w_{lk}^P}{\partial w_{lk}^I} = \cos(\varphi_l - \theta_k). \quad (33)$$

Consequently, the derivatives for the weight correction rule based on the gradient descent method are summarized as follows:

$$\frac{\partial E}{\partial w_{lk}^R} = -\sin(\theta_l - \hat{\theta}_l) f'_p(\varphi_l) \sin(\varphi_l - \theta_k) / u_l, \quad (34)$$

$$\frac{\partial E}{\partial w_{lk}^I} = \sin(\theta_l - \hat{\theta}_l) f'_p(\varphi_l) \cos(\varphi_l - \theta_k) / u_l, \quad (35)$$

for $k = 0, 1, \dots, N_k$ and $l = 1, 2, \dots, N_l$. For the calculation of the derivatives with respect to the bias parameters, w_{l0}^R and w_{l0}^I , we take $\theta_0 = 0$. By combining the derivatives in Eqs. (34)-(35), we get the weight correction rule as follows:

$$\tilde{w}_{lk} = w_{lk} - \eta \sin(\theta_l - \hat{\theta}_l) f'_p(\varphi_l) \exp(\varphi_l - \theta_k + \pi/2) / u_l. \quad (36)$$

4.2 Backpropagation Algorithm

Next we consider backwards propagation of the errors. After corrections of the weight coefficients in the output layer neurons, the weights in the hidden layer neurons are updated. Suppose that the weight coefficient between the k th hidden unit and the j th input unit is updated as follows:

$$\tilde{w}_{kj} = w_{kj} + \Delta w_{kj}, \quad (37)$$

where $w_{kj} = w_{kj}^R + iw_{kj}^I$ and $\tilde{w}_{kj} = \tilde{w}_{kj}^R + i\tilde{w}_{kj}^I$ are the current and updated weights, respectively. The variations are obtained with gradient descent by

$$\Delta w_{kj}^R = -\eta \frac{\partial E}{\partial w_{kj}^R}, \quad (38)$$

$$\Delta w_{kj}^I = -\eta \frac{\partial E}{\partial w_{kj}^I}, \quad (39)$$

where η is the learning rate.

The derivatives in the righthand sides are given by the chain rule as follows:

$$\frac{\partial E}{\partial w_{kj}^R} = \left(\sum_{l=1}^{N_l} \frac{\partial E}{\partial \theta_l} \frac{d\theta_l}{d\varphi_l} \frac{\partial \varphi_l}{\partial \theta_k} \right) \frac{d\theta_k}{d\varphi_k} \frac{\partial \varphi_k}{\partial w_{kj}^R} \frac{\partial w_{kj}^P}{\partial w_{kj}^R}, \quad (40)$$

$$\frac{\partial E}{\partial w_{kj}^I} = \left(\sum_{l=1}^{N_l} \frac{\partial E}{\partial \theta_l} \frac{d\theta_l}{d\varphi_l} \frac{\partial \varphi_l}{\partial \theta_k} \right) \frac{d\theta_k}{d\varphi_k} \frac{\partial \varphi_k}{\partial w_{kj}^I} \frac{\partial w_{kj}^P}{\partial w_{kj}^I}. \quad (41)$$

Most terms in the righthand sides of the above equations, except for $\partial \varphi_l / \partial \theta_k$, can be calculated as in the previous subsection. To calculate $\partial \varphi_l / \partial \theta_k$, we focus on the following equation:

$$u_l e^{i\varphi_l} = \sum_{k=1}^{N_k} w_{lk} e^{i\theta_k} = \left(\sum_{k=1}^{N_k} w_{lk}^R \cos \theta_k \right) + i \left(\sum_{k=1}^{N_k} w_{lk}^I \sin \theta_k \right),$$

which leads to

$$\tan \varphi_l = \frac{u_l \sin \varphi_l}{u_l \cos \varphi_l} = \frac{\sum_k w_{lk}^I \sin \theta_k}{\sum_k w_{lk}^R \cos \theta_k}.$$

By differentiating both sides with respect to θ_k , we obtain

$$\frac{1}{\cos^2 \varphi_l} \frac{\partial \varphi_l}{\partial \theta_k} = \frac{(w_{lk}^I \cos \theta_k)(u_l \cos \varphi_l) + (u_l \sin \varphi_l)(w_{lk}^R \sin \theta_k)}{u_l^2 \cos^2 \varphi_l}.$$

Hence,

$$\frac{\partial \varphi_l}{\partial \theta_k} = \frac{w_{lk}^R \sin \theta_k \sin \varphi_l + w_{lk}^I \cos \theta_k \cos \varphi_l}{u_l}. \tag{42}$$

Consequently, we obtain the error backpropagation formula for the three-layer network in Eqs. (40)-(41). Even when the number of layers is more than three, the procedure of the backpropagation algorithm is almost the same.

5 Simulation Results

A popular task to evaluate the performance of a learning algorithm for feed-forward neural networks is function approximation. In this section, M pairs of input and output data are randomly generated by the following function:

$$y = h(x) + \xi \pmod{2\pi}. \tag{43}$$

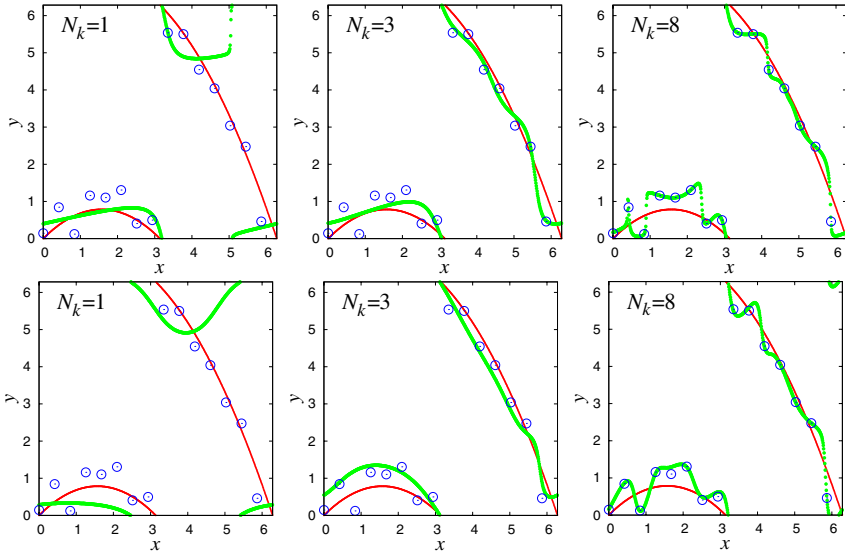


Fig. 2. The original function $y = h(x)$ (thin) and the functions (thick) approximated by three-layer networks. The number of training data is $M = 15$. The number of hidden units is indicated by N_k . (Upper) The proposed backpropagation algorithm based on gradient descent method. (Lower) Backpropagation algorithm based on linear correction rule.

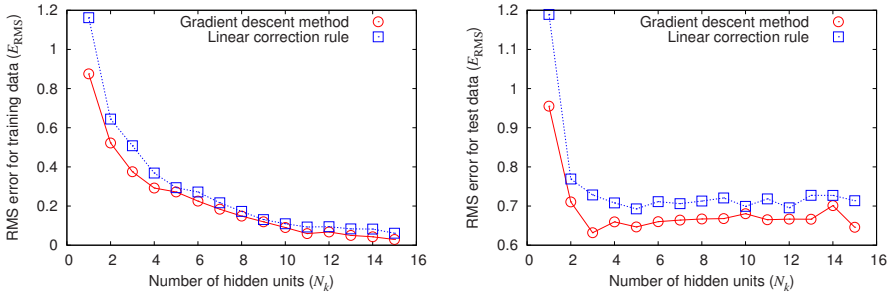


Fig. 3. (Left) The RMS error for the training data with variation of the number of hidden units. (Right) The RMS error for the test data with variation of the number of hidden units.

where $h(x) = x(\pi - x)/\pi$ and ξ indicates white Gaussian noise with mean zero and variance $\sigma = 0.25$. The aim is to approximate $y = h(x)$ using the training data set. For comparison, the same experiments are conducted with backpropagation algorithm based on linear correction rule [14]. Figure 2 shows the results of function approximation task for different numbers of hidden units in the three-layer network. The number of training data indicated by the circles is $M = 15$. The number of iterations of weight corrections is fixed at 10000. In both methods, the activation function (7) is used. When $N_k = 1$, the approximated function is far from the original function. At $N_k = 3$, the best approximation is achieved in both methods. A further increase of N_k leads to overfitting of the training data as exemplified in the case of $N_k = 8$.

Figure 3 shows the root-mean-square (RMS) error for the training and test data. The mean values of 100 trials are plotted. The M test data are newly generated by (43). The RMS error for the training data is almost monotonically decreasing with increase of the hidden units in both methods. The RMS error for the test data with the proposed method is smaller than that with the linear correction method. It suggests that our method has a better generalization ability.

6 Conclusions

We have presented a backpropagation learning algorithm based on a gradient descent method for multilayer feedforward phasor neural networks. The gradient descent method relies on the continuous and differentiable activation function. In order to demonstrate the performance of the proposed method, we have conducted numerical simulations on function approximation task. In comparison with the linear correction method, the proposed gradient descent method can be advantageous in terms of generalization ability. The learning method for phasor neural networks can be useful for processing various information with periodicity or circularity.

Acknowledgments. This work was partially supported by Grant-in-Aid for Young Scientist (B) from the Ministry of Education, Culture, Sports, Science and Technology of Japan (No.19700214).

References

1. Mandic, D., Goh, V. (eds.): *Complex Valued Nonlinear Adaptive Filters*. John Wiley & Sons, Ltd., Chichester (2009)
2. Hirose, A. (ed.): *Complex-Valued Neural Networks: Theories and Applications*. Innovative Intelligence, vol. 5. World Scientific, Singapore (2004)
3. Nitta, T. (ed.): *Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters*. Information Science Reference, Pennsylvania (2009)
4. Nitta, T.: An extension of the back-propagation algorithm to complex numbers. *Neural Networks* 10, 1391–1415 (1997)
5. Georgiou, G.M., Koutsougeras, C.: Complex domain backpropagation. *IEEE Trans. CAS-II* 39(5), 330–334 (1992)
6. Hirose, A.: Continuous complex-valued back-propagation learning. *Electronics Lett.* 28(20), 1854–1855 (1992)
7. Noest, A.J.: Associative memory in sparse phasor neural networks. *Europhys. Lett.* 6(6), 469–474 (1988)
8. Noest, A.J.: Discrete-state phasor neural networks. *Phys. Rev. A* 38, 2196–2199 (1988)
9. Jankowski, S., Lozowski, A., Zurada, J.M.: Complex-valued multistate neural associative memory. *IEEE Trans. Neural Netw.* 7, 1491–1496 (1996)
10. Müezzinoğlu, M.K., Güzeliş, C., Zurada, J.M.: A new design method for the complex-valued multistate hopfield associative memory. *IEEE Trans. Neural Netw.* 14(4), 891–899 (2003)
11. Lee, D.-L.: Improvements of complex-valued hopfield associative memory by using generalized projection rules. *IEEE Trans. Neural Netw.* 17(5), 1341–1347 (2006)
12. Aizenberg, I., Aizenberg, N., Vandewalle, J. (eds.): *Multi-Valued and Universal Binary Neurons: Theory, Learning, and Applications*. Kluwer Academic Publishers, Dordrecht (2000)
13. Tanaka, G., Aihara, K.: Complex-Valued Multistate Associative Memory with Nonlinear Multilevel Functions for Gray-Level Image Reconstruction. *IEEE Trans. Neural Netw.* 20(9), 1463–1473 (2009)
14. Aizenberg, I., Moraga, C.: Multilayer feedforward neural network based on multi-valued neurons (mlmvm) and a backpropagation learning algorithm. *Soft Comput.* 11, 169–183 (2007)
15. Si, J., Michel, A.N.: Analysis and synthesis of discrete-time neural networks with multilevel threshold functions. *IEEE Trans. Neural Netw.* 6(1), 105–116 (1995)
16. Zurada, J.M., Cloete, I., van der Poel, E.: Generalized hopfield networks with multiple stable states. *Neurocomput.* 13, 135–149 (1996)

SNIWD: Simultaneous Weight Noise Injection with Weight Decay for MLP Training

John Sum¹ and Kevin Ho²

¹ Institute of Technology Management, National Chung Hsing University
Taichung 402, Taiwan
pfsun@nchu.edu.tw

² Department of Computer Science and Communication Engineering,
Providence University, Sha-Lu, Taiwan
ho@pu.edu.tw

Abstract. Despite noise injecting during training has been demonstrated with success in enhancing the fault tolerance of neural network, theoretical analysis on the dynamic of this noise injection-based online learning algorithm has far from complete. In particular, the convergence proofs for those algorithms have not been shown. In this regards, this paper presents an empirical study on the non-convergence properties of injecting weight noises during training a multilayer perceptron, and an online learning algorithm called SNIWD (simultaneous noise injection and weight decay) to overcome such non-convergence problem. Simulation results show that SNIWD is able to improve the convergence and enforce small magnitude on the network parameters (input weights, input biases and output weights). Moreover, SNIWD is able to make the network have similar fault tolerance ability as using pure noise injection approach.

1 Introduction

Improve tolerance of a neural network towards random node fault, stuck-at node fault and weight noise have been researching for almost two decades [4,6,5,7,9,12,13,16,17,19]. Many methods such as injecting random node fault [18,3], injecting weight noise during training (for multilayer perceptrons (MLP) [14,15], a recurrent neural network (RNN) [11], or a pulse-coupled neural networks (PCNN) [8]) or node noise (response variability) during training [2] (for PCNN) during training have been developed and demonstrated with success via intensive computer simulations. Despite the idea of injecting weight noise during training is straight forward and its implementation is extremely elegant, theoretical analysis regarding their convergence and the objective functions in which the algorithms are minimizing is scarce [12,14,15].

Murray and Edward although have found that injecting multiplicative weight noise can enhance the fault tolerance of a MLP [15], they have not put forward the objective function for this algorithm. While G.An in [1] has attempted to derive an objective function for injecting weight-noise during training (see Section

4 in [1]), he failed to prove the convergence of this algorithm and nevertheless the objective function derived is not the true one. In terms of Murray & Edward’s terminology, the objective derived by G.An is essentially the prediction error of a MLP if weight noise is injected after training. Until very recent, Ho *et al* [10] have shown the first complete analysis on the convergence of injecting output weight noise (either multiplicative or additive) during training a radial basis function (RBF) network.

In view of lacking understand on injecting weight noise during training a MLP, simulated experiments have been conducted. We found that pure noise injection during training might lead to non-convergence of network parameters, even the training error has been converged. Rather, adding weight decay together with noise injection during training is able to overcome such non-convergence problem. In this paper, we will present this comparative study based on purely noise injection training algorithm and simultaneous weight noise injection with weight decay (SINWD).

In the next section, the online weight noise injection algorithms will be presented. Their convergence properties, in terms of training error and network parameters, and their fault tolerance abilities will be shown by a simple example in Section 3. Section 4 gives the conclusions of this paper.

2 Noise Injection during Training

Let $\mathbf{f}(\cdot, \cdot) \in R^l$ be a single output multilayer perceptron (MLP) consisting of m hidden nodes, n input nodes and l linear output nodes.

$$\mathbf{f}(\mathbf{x}, \mathbf{w}) = \mathbf{D}^T \mathbf{z}(\mathbf{A}^T \mathbf{x} + \mathbf{c}), \quad (1)$$

where $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_l] \in R^{m \times l}$ is the hidden to output weight vector, $\mathbf{z} = (z_1, z_2, \dots, z_m)^T \in R^m$ is the output of the hidden nodes, $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m] \in R^{n \times m}$ is the input to hidden weight matrix, $\mathbf{a}_i \in R^n$ is the input weight vector of the i^{th} hidden node and $\mathbf{c} \in R^m$ is the input to hidden bias vector.

\mathbf{w} in [1] is a vector augmenting all the parameters, i.e.

$$\mathbf{w} = (\mathbf{d}_1^T, \mathbf{d}_2^T, \dots, \mathbf{d}_l^T, \mathbf{a}_1^T, \mathbf{a}_2^T, \dots, \mathbf{a}_m^T, \mathbf{c}^T)^T.$$

For $i = 1, 2, \dots, m$, $z_i(\mathbf{x}, \mathbf{a}_i, c_i) = \tau(\mathbf{a}_i^T \mathbf{x} + c_i)$, where $\tau(\cdot)$ is the neuronal transfer function. Training dataset is denoted by $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^N$. The random noise vector is denoted by \mathbf{b} . For simplicity, we assume that there is only one output node, i.e. $l = 1$. In such case, the gradient of $f(\mathbf{x}, \mathbf{w})$ with respect to \mathbf{w} is denoted by $g(\mathbf{x}_t, \mathbf{w}(t))$. The Hessian matrix of $f(\mathbf{x}, \mathbf{w})$ is denoted by $g_{\mathbf{w}}(\mathbf{x}_t, \mathbf{w}(t))$.

Table 1. Settings of the experiments

	Pure noise injection	With weight decay
Add. weight noise (Fig 1)	$S_b = .01, \alpha = 0$	$S_b = .01, \alpha = .00001$
Mul. weight noise (Fig 3)	$S_b = .01, \alpha = 0$	$S_b = .01, \alpha = .00001$

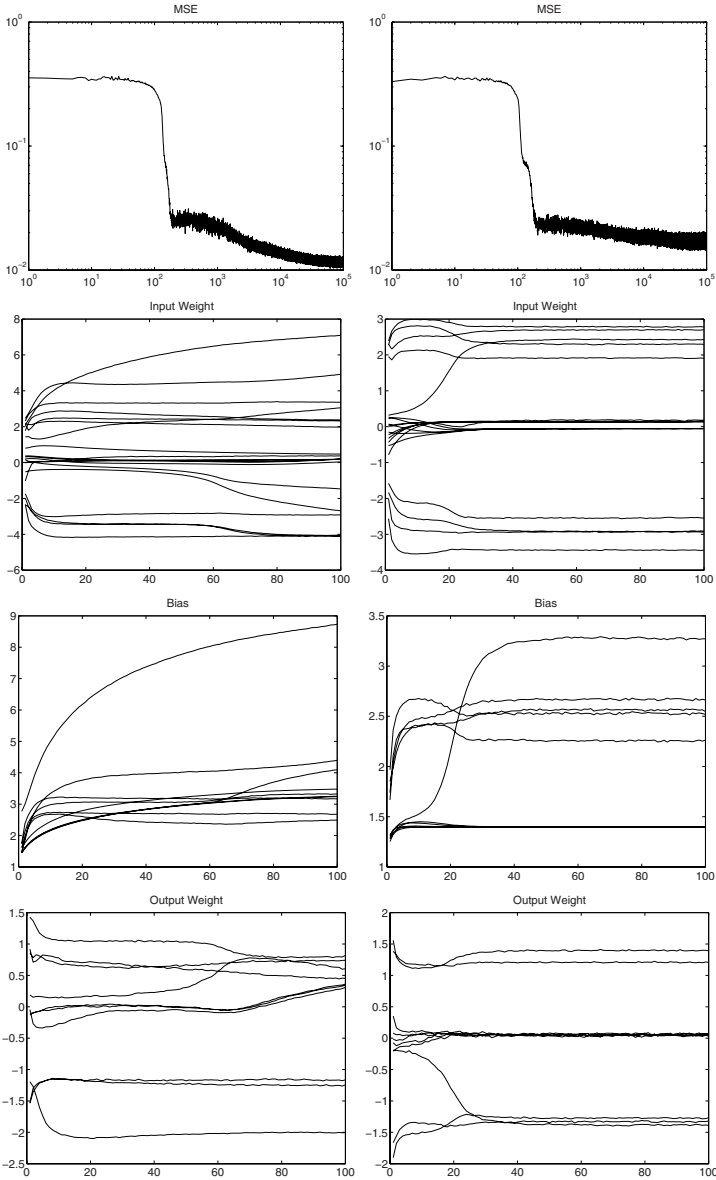


Fig. 1. Dynamical changes of the network parameters while additive weight noise is injected during training. Note that the total number of training steps is 100×1000 . Every two consecutive points are taken at an interval of 1000 steps.

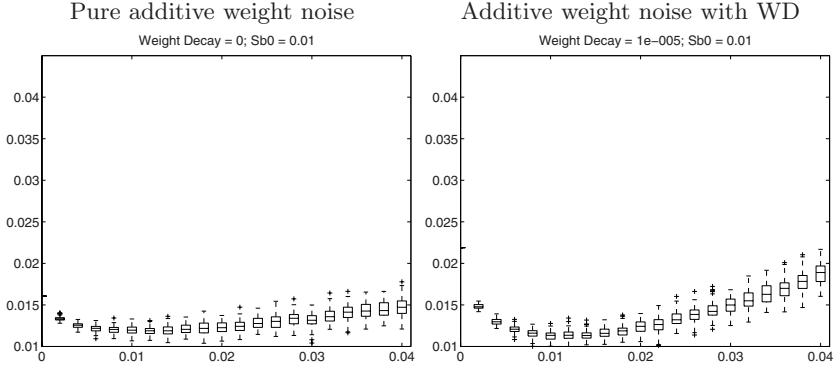


Fig. 2. Testing error versus different values of S'_b , for the networks obtained in Figure [1](#)

2.1 Pure Weight Noise Injection

The online **weight noise injection** training algorithm for $f(\mathbf{x}, \mathbf{w})$ given a dataset \mathcal{D} can be written as follows :

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mu_t (y_t - f(\mathbf{x}_t, \tilde{\mathbf{w}}(t))) \mathbf{g}(\mathbf{x}_t, \tilde{\mathbf{w}}(t)). \quad (2)$$

$$\tilde{\mathbf{w}}(t) = \mathbf{w}(t) + \mathbf{b} \odot \mathbf{w}(t). \quad (\text{multiplicative weight noise}) \quad (3)$$

$$\tilde{\mathbf{w}}(t) = \mathbf{w}(t) + \mathbf{b}. \quad (\text{additive weight noise}) \quad (4)$$

Here $\mathbf{b} \odot \mathbf{w} = (b_1 w_1, b_2 w_2, \dots, b_M w_M)^T$ and b_i , for all i , is a mean zero Gaussian distribution with variance S_b .

2.2 SNIWD

For simultaneous weight noise injection and weight decay (SNIWD), the update equations are similar except the decay term is added.

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mu_t \{ (y_t - f(\mathbf{x}_t, \tilde{\mathbf{w}}(t))) \mathbf{g}(\mathbf{x}_t, \tilde{\mathbf{w}}(t)) - \alpha \mathbf{w}(t) \}. \quad (5)$$

$$\tilde{\mathbf{w}}(t) = \mathbf{w}(t) + \mathbf{b} \odot \mathbf{w}(t). \quad (\text{multiplicative weight noise}) \quad (6)$$

$$\tilde{\mathbf{w}}(t) = \mathbf{w}(t) + \mathbf{b}. \quad (\text{additive weight noise}) \quad (7)$$

Clearly, the difference between pure noise injection during training, and the one with weight decay lies in the last term of the update equation, i.e. $-\alpha \mathbf{w}(t)$, which can limit the growth of $\|\mathbf{w}(t)\|$ to infinity.

3 Simulation Study

To illustrate the effect of injection noise during training MLP with and without adding weight decay, a training dataset consisting of 100 samples that are generated from an XOR function is used for the MLP training.

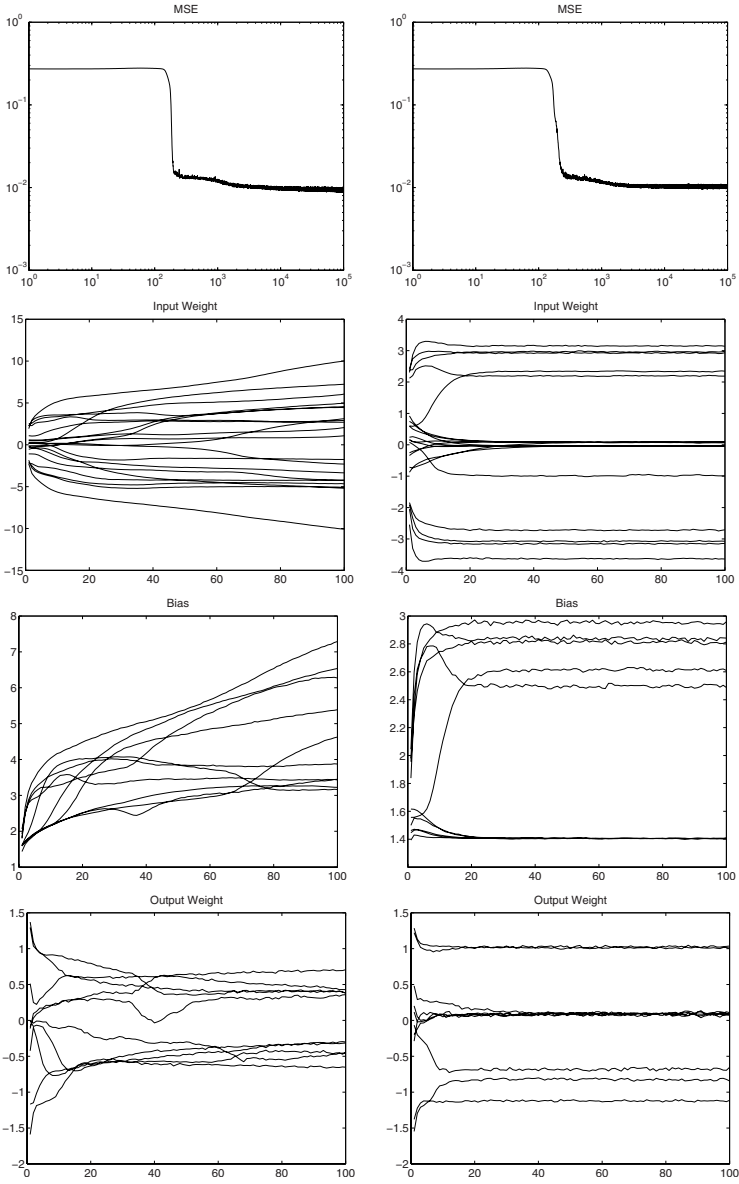


Fig. 3. Dynamical changes of the network parameters while additive weight noise is injected during training. Note that the total number of training steps is 100×1000 . Every two consecutive points are taken at an interval of 1000 steps.

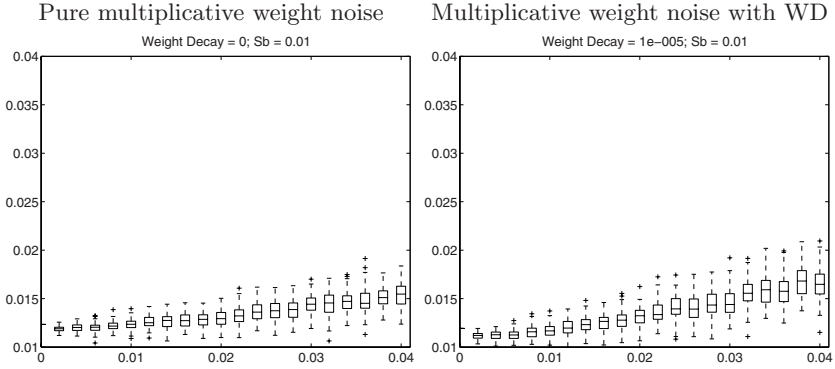


Fig. 4. Testing error versus different values of S'_b , for the networks obtained in Figure 3

Let (\mathbf{x}_k, y_k) be the k^{th} input and output pair. \mathbf{x}_k is uniformly random selected from $[-1, 1] \times [-1, 1]$. The output vector $y_k \in \{0, 1\}$ is generated by the following equation.

$$y_k = \mathbf{sign}(x_{k1})\mathbf{sign}(x_{k2}). \quad (8)$$

Then, an MLP consisting of 2 input nodes, 10 hidden nodes and 1 linear output nodes is trained. Four experiments are carried out. The values of S_b and α are depicted in Table 1. The step size for all eight experiments is set to 0.05. The change of parameters during training are shown in Figure 1 and Figure 3. To validate the fault tolerance ability, each network that is trained with online additive (multiplicative) weight noise injection will be injected S'_b additive (multiplicative) weight noise after training and then the testing error is evaluated. The last step is repeated 100 times, the statistics of the testing errors are displayed in box-plot form and shown in Figure 2 and Figure 4 respectively for additive weight noise injection and multiplicative weight noise injection. The range of S'_b is from 0 to 0.04.

In accordance with the simulation results, it is clear that the network parameters do not converge for pure weight noise injection cases. Even the training error has shown converge, many network parameters are still increasing. Adding weight decay is able to control the growth of the network parameters, especially the input weights. If weight decay is added, their magnitudes converge to below 4. Without weight decay, their magnitude can diverge to as large as 10, see Figure 3.

Moreover, as observed from Figure 2 and Figure 4 that the fault tolerance abilities of a network trained by pure noise injection and SNIWD are quite similar. Except that, SNIWD gives slightly better performance when S'_b is close to 0.01. For S'_b is larger than 0.02, the situation is reverse.

4 Conclusion

In this paper, we have presented simulation results comparing the convergence of network parameters (including input weights, input biases and output weights)

that are obtained by purely noise injection and simultaneous noise injection with weight decay. We have found that purely injecting weight noise during training a MLP might not be able to improve its fault tolerance, as the some of network parameters might diverge. By simulations, we have found that adding weight decay simultaneously with weight noise injection during training is able to overcome such problem. For a network that is trained by SNIWD approach, its network parameters are with smaller magnitude compared with pure weight noise injection approach. Convergence of network parameters is almost guaranteed. The fault tolerance ability of that network is comparable to that is trained by purely noise injection approach. Due to page limit, we are not able to derive the objective functions in which those algorithms are minimizing in this paper. Those theoretical results will be presented in our future papers.

Acknowledgement

The research work reported in this paper is supported in part by Taiwan NSC Research Grant 97-2221-E-005-050.

References

1. An, G.: The effects of adding noise during backpropagation training on a generalization performance. *Neural Computation* 8, 643–674 (1996)
2. Basalyga, G., Salinas, E.: When response variability increases neural network robustness to synaptic noise. *Neural Computation* 18, 1349–1379 (2006)
3. Bolt, G.: Fault tolerant in multi-layer Perceptrons. PhD Thesis, University of York, UK (1992)
4. Bernier, J.L., et al.: Obtaining fault tolerance multilayer perceptrons using an explicit regularization. *Neural Processing Letters* 12, 107–113 (2000)
5. Cavalieri, S., Mirabella, O.: A novel learning algorithm which improves the partial fault tolerance of multilayer NNs. *Neural Networks* 12, 91–106 (1999)
6. Chiu, C.T., et al.: Modifying training algorithms for improved fault tolerance. In: *ICNN 1994*, vol. I, pp. 333–338 (1994)
7. Deodhare, D., Vidyasagar, M., Sathiya Keerthi, S.: Synthesis of fault-tolerant feed-forward neural networks using minimax optimization. *IEEE Transactions on Neural Networks* 9(5), 891–900 (1998)
8. Edwards, P.J., Murray, A.F.: Fault tolerant via weight noise in analog VLSI implementations of MLP's – A case study with EPSILON. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 45(9), 1255–1262 (1998)
9. Hammadi, N.C., Hideo, I.: A learning algorithm for fault tolerant feedforward neural networks. *IEICE Transactions on Information & Systems* E80-D(1) (1997)
10. Ho, K., Leung, C.S., Sum, J.: On weight-noise-injection training. In: Koeppen, M., Kasabov, N., Coghill, G. (eds.) *Advances in Neuro Information Processing*. LNCS, vol. 5507, pp. 919–926. Springer, Heidelberg (2009)
11. Jim, K.C., Giles, C.L., Horne, B.G.: An analysis of noise in recurrent neural networks: Convergence and generalization. *IEEE Transactions on Neural Networks* 7, 1424–1438 (1996)

12. Kamiura, N., et al.: On a weight limit approach for enhancing fault tolerance of feedforward neural networks. *IEICE Transactions on Information & Systems* E83-D(11) (2000)
13. Leung, C.S., Sum, J.: A fault tolerant regularizer for RBF networks. *IEEE Transactions on Neural Networks* 19(3), 493–507 (2008)
14. Murray, A.F., Edwards, P.J.: Synaptic weight noise during multilayer perceptron training: fault tolerance and training improvements. *IEEE Transactions on Neural Networks* 4(4), 722–725 (1993)
15. Murray, A.F., Edwards, P.J.: Enhanced MLP performance and fault tolerance resulting from synaptic weight noise during training. *IEEE Transactions on Neural Networks* 5(5), 792–802 (1994)
16. Neti, C., Schneider, M.H., Young, E.D.: Maximally fault tolerance neural networks. *IEEE Transactions on Neural Networks* 3(1), 14–23 (1992)
17. Phatak, D.S., Koren, I.: Complete and partial fault tolerance of feedforward neural nets. *IEEE Transactions on Neural Networks* 6, 446–456 (1995)
18. Sequin, C.H., Clay, R.D.: Fault tolerance in feedforward artificial neural networks. *Neural Networks* 4, 111–141 (1991)
19. Sum, J., Leung, C.S., Ho, K.: On objective function, regularizer and prediction error of a learning algorithm for dealing with multiplicative weight noise. *IEEE Transactions on Neural Networks* 20(1) (January 2009)

Tracking in Reinforcement Learning

Matthieu Geist^{1,2,3}, Olivier Pietquin¹, and Gabriel Fricout²

¹ IMS Research Group, Supélec, Metz, France

² MC Cluster, ArcelorMittal Research, Maizières-lès-Metz, France

³ CORIDA project-team, INRIA Nancy - Grand Est, France

Abstract. Reinforcement learning induces non-stationarity at several levels. Adaptation to non-stationary environments is of course a desired feature of a fair RL algorithm. Yet, even if the environment of the learning agent can be considered as stationary, generalized policy iteration frameworks, because of the interleaving of learning and control, will produce non-stationarity of the evaluated policy and so of its value function. Tracking the optimal solution instead of trying to converge to it is therefore preferable. In this paper, we propose to handle this tracking issue with a Kalman-based temporal difference framework. Complexity and convergence analysis are studied. Empirical investigations of its ability to handle non-stationarity is finally provided.

Keywords: Reinforcement learning, value function approximation, tracking, Kalman filtering.

1 Introduction

Reinforcement learning (RL) [1] is a general paradigm in which an agent learns to control a dynamic system (its *environment*) through examples of real interactions without any model of the physics ruling this system. A feedback signal is observed by this agent after each interaction as a reward information, which is a local hint about the quality of the control. When addressing a reinforcement learning problem, one considers the system as made up of states and accepting actions from the controlling agent. The objective of the agent is to learn the mapping from states to actions (a *policy*) that maximizes the expected cumulative reward over the long term, which it locally models as a so-called value or *Q*-function. Reinforcement learning induces non-stationarity at several levels. First, as in a lot of real-world machine learning applications, adaptation to non-stationary environments is a desired feature of a learning method. Yet most of existing machine learning algorithms assume stationarity of the problem and aim at converging to a fixed solution. Few attempts to handle non-stationarity of the environment in RL can be found in the litterature. Most of them are based on interleaving of RL and planning such as in the Dyna-Q algorithm [2]. Tracking value function is proposed by [3], which can be seen as a specific case of the proposed approach. Second, a large class of RL approaches consists in alternatively learning the value function of a given policy, and then improving the policy according to the learnt values. This is known as *generalized policy iteration* [4]. This scheme suggests to have a value function learner. However, because of the policy improvement phase, the value function changes together with the

policy and makes it non-stationary. In both cases, tracking the value function rather than converging to it seems preferable. Other arguments can be discussed on the advantages of tracking *vs* converging even in stationary environments [3]. To address this issue, we propose a statistical approach to value function approximation in RL based on Kalman filtering, namely the *Kalman Temporal Difference* framework. Kalman filtering is indeed an efficient solution to tracking problems and is shown here to apply positively to the problem at sight. Readers are invited to refer to [4] for a deeper theoretical description. Contributions of this paper are the analysis of this framework (computational complexity, bias caused by stochastic transitions and convergence) and a set of experimental results which show its ability to handle non-stationary (for a non-stationary system and in the case of interlacing of control and learning), as well as sample efficiency.

2 Background

Originally, Kalman filtering [5] aims at online tracking the hidden state of a non-stationary dynamic system through indirect observations of this state. The idea behind KTD is to express the problem of value function approximation in RL as a filtering problem. Considering a parametric value function approximator, the parameters are the hidden state to be tracked, the observation being the reward linked to the parameters through a Bellman equation.

2.1 Reinforcement Learning

This paper is placed in the framework of Markov decision process (MDP). An MDP is a tuple $\{S, A, P, R, \gamma\}$, where S is the state space, A the action space, $P : s, a \in S \times A \rightarrow p(\cdot | s, a) \in \mathcal{P}(S)$ a family of transition probabilities, $R : S \times A \times S \rightarrow \mathbb{R}$ the bounded reward function, and γ the discount factor. A policy π associates to each state a probability over actions, $\pi : s \in S \rightarrow \pi(\cdot | s) \in \mathcal{P}(A)$. The value function of a given policy is defined as $V^\pi(s) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, \pi]$ where r_i is the immediate reward observed at time step i , and the expectation is done over all possible trajectories starting in s given the system dynamics and the followed policy. The Q -function allows a supplementary degree of freedom for the first action and is defined as $Q^\pi(s, a) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, a_0 = a, \pi]$. RL aims at finding (through interactions) the policy π^* which maximises the value function for every state: $\pi^* = \operatorname{argmax}_\pi (V^\pi)$. Two schemes among others can lead to the optimal policy. First, *policy iteration* implies learning the value function of a given policy and then improving the policy, the new one being greedy respectively to the learned value function. It requires solving the *Bellman evaluation equation*, which is given here for the value and Q -functions:

$$V^\pi(s) = E_{s', a | \pi, s} [R(s, a, s') + \gamma V^\pi(s')], \forall s \quad (1)$$

$$Q^\pi(s, a) = E_{s', a' | \pi, s, a} [R(s, a, s') + \gamma Q^\pi(s', a')], \forall s, a \quad (2)$$

The second scheme, *value iteration*, aims directly at finding the optimal policy. It requires solving the *Bellman optimality equation*:

$$Q^*(s, a) = E_{s' | s, a} [R(s, a, s') + \gamma \max_{b \in A} Q^*(s', b)], \forall s, a \quad (3)$$

2.2 Kalman Temporal Differences

For the sake of generality, the following notations are adopted, given that the aim is respectively the value or the Q -function evaluation or the Q -function optimization:

$$t_i = \begin{cases} (s_i, s_{i+1}) \\ (s_i, a_i, s_{i+1}, a_{i+1}) \\ (s_i, a_i, s_{i+1}) \end{cases} \quad \text{and} \quad g_{t_i}(\theta_i) = \begin{cases} \hat{V}_{\theta_i}(s_i) - \gamma \hat{V}_{\theta_i}(s_{i+1}) \\ \hat{Q}_{\theta_i}(s_i, a_i) - \gamma \hat{Q}_{\theta_i}(s_{i+1}, a_{i+1}) \\ \hat{Q}_{\theta_i}(s_i, a_i) - \gamma \max_b \hat{Q}_{\theta_i}(s_{i+1}, b) \end{cases} \quad (4)$$

where \hat{V}_{θ} (resp. \hat{Q}_{θ}) is a parametric representation of the value (resp. Q -) function and θ is the parameter vector. A statistical point of view is adopted and the problem at sight is stated in a so-called *state-space formulation* (that is the value function approximation problem is cast into the Kalman filtering paradigm):

$$\begin{cases} \theta_i = \theta_{i-1} + v_i & \text{(evolution equation)} \\ r_i = g_{t_i}(\theta_i) + n_i & \text{(observation equation)} \end{cases} \quad (5)$$

The first equation (*evolution equation*), is the key of non-stationarity handling. It specifies that the parameter vector evolves with time according to a random walk. The random walk model is chosen for its simplicity. Expectation of θ_i corresponds to the optimal estimation of the value function at time step i . The evolution noise v_i is centered, white, independent and of variance matrix P_{v_i} (to be chosen by the practitioner). The second equation (*observation equation*) links the observed transition to the value (or Q -) function through one of the Bellman equations. The observation noise n_i is supposed centered, white, independent and of variance P_{n_i} (also to be chosen by the practitioner). Notice that this white noise assumption does not hold for stochastic MDP (see Sec. 3.2). KTD is a second order algorithm (and thus sample efficient): it updates the mean parameter vector, but also the associated variance matrix. It breaks down in three steps (see also algorithm I). First, the *prediction* step (*i*) consists in predicting the parameters mean and covariance at time step i according to the evolution equation and using previous estimates. Then some *statistics of interest* are computed (*ii*). Third, the *correction* step (*iii*) consists in correcting first and second order moments of the parameter random vector according to the Kalman gain K_i (obtained thanks to statistics computed at step (*ii*)), the predicted reward $\hat{r}_{i|i-1}$ and the observed reward r_i (the difference between the two being a form of temporal difference error). The statistics of interest are generally not analytically computable, except in the linear case, which does not hold for nonlinear parameterization and for the Bellman optimality equation, because of the max operator. Yet, a derivative-free approximation scheme, the *unscented transform* (UT) [6], is used to estimate first and second order moments of nonlinearly mapped random vectors. Let X be a random vector of size n and $Y = f(X)$ its nonlinear mapping. A set of $2n + 1$ so-called sigma-points is computed as follows:

$$\begin{cases} x^{(0)} = \bar{X} & w_0 = \frac{\kappa}{n+\kappa}, \quad j = 0 \\ x^{(j)} = \bar{X} + (\sqrt{(n+\kappa)P_X})_j & w_j = \frac{1}{2(n+\kappa)}, \quad 1 \leq j \leq n \\ x^{(j)} = \bar{X} - (\sqrt{(n+\kappa)P_X})_{n-j} & w_j = \frac{1}{2(n+\kappa)}, \quad n+1 \leq j \leq 2n \end{cases} \quad (6)$$

where \bar{X} is the mean of X , P_X is its variance matrix, κ is a scaling factor which controls the accuracy [6], and $(\sqrt{P_X})_j$ is the j^{th} column of the Cholesky decomposition of P_X . Then the image through the mapping f is computed for each of these sigma-points:

$$y^{(j)} = f(x^{(j)}), \quad 0 \leq j \leq 2n \quad (7)$$

The set of sigma-points and their images can then be used to compute the following approximations:

$$\begin{cases} \bar{Y} \approx \bar{y} = \sum_{j=0}^{2n} w_j y^{(j)} \\ P_Y \approx \sum_{j=0}^{2n} w_j (y^{(j)} - \bar{y})(y^{(j)} - \bar{y})^T \\ P_{XY} \approx \sum_{j=0}^{2n} w_j (x^{(j)} - \bar{X})(y^{(j)} - \bar{y})^T \end{cases} \quad (8)$$

Using the UT practical algorithms can be derived. At time-step i , a set of sigma-points is computed from predicted random parameters characterized by mean $\hat{\theta}_{i|i-1}$ and variance $P_{i|i-1}$. Predicted rewards are then computed as images of these sigma-points using one of the observation functions (4). Then sigma-points and their images are used to compute statistics of interest. This gives rise to three algorithms, namely KTD-V, KTD-SARSA and KTD-Q, given that the aim is to evaluate the value or Q -function of a given policy or directly the optimal Q -function. They are summarized in Algorithm 1, p being the number of parameters.

3 KTD Analysis

3.1 Computational Cost

The UT involves a Cholesky decomposition, which can be performed in $O(p^2)$ instead of $O(p^3)$ when done with a square-root approach [7]. The different algorithms imply to evaluate $2p + 1$ times the g_{t_i} function at each time-step. For KTD-V or KTD-SARSA and a general parameterization, each evaluation is bounded by $O(p)$. For KTD-Q, the maximum over actions has to be computed. Let \mathcal{A} be the cardinality of action space if finite, the computational complexity of the algorithm used to search the maximum otherwise (e.g., the number of samples for Monte Carlo). Then each evaluation is bounded by $O(p\mathcal{A})$. The rest of operations is basic linear algebra, and is bounded by $O(p^2)$. Thus the global computational complexity (per iteration) of KTD-V and KTD-SARSA is $O(p^2)$, and KTD-Q is in $O(\mathcal{A}p^2)$. This is comparable to approaches such as LSTD [8] (nevertheless with the additional ability to handle nonlinear parameterization).

3.2 Stochastic MDP

The KTD framework assumes a white observation noise. In the case of deterministic MDP, this observation noise only models the inductive bias introduced by function approximation. But in stochastic MDP, this noise includes the stochasticity of transitions as well and cannot be considered white anymore. Similarly to other second

Algorithm 1. KTD-V, KTD-SARSA and KTD-Q*Initialization;*priors $\hat{\theta}_{0|0}$ and $P_{0|0}$;**for** $i \leftarrow 1, 2, \dots$ **do**

Observe transition $t_i = \begin{cases} (s_i, s_{i+1}) \text{ (KTD-V)} \\ (s_i, a_i, s_{i+1}, a_{i+1}) \text{ (KTD-SARSA)} \\ (s_i, a_i, s_{i+1}) \text{ (KTD-Q)} \end{cases}$ and reward r_i ;

Prediction Step;

$$\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1};$$

$$P_{i|i-1} = P_{i-1|i-1} + P_{v_i};$$

Sigma-points computation ;

$$\Theta_{i|i-1} = \left\{ \hat{\theta}_{i|i-1}^{(j)}, \quad 0 \leq j \leq 2p \right\} \text{ (using the UT, from } \hat{\theta}_{i|i-1} \text{ and } P_{i|i-1});$$

$$\mathcal{W} = \{w_j, \quad 0 \leq j \leq 2p \};$$

$$\mathcal{R}_{i|i-1} =$$

$$\begin{cases} \left\{ \hat{r}_{i|i-1}^{(j)} = \hat{V}_{\hat{\theta}_{i|i-1}^{(j)}}(s_i) - \gamma \hat{V}_{\hat{\theta}_{i|i-1}^{(j)}}(s_{i+1}), \quad 0 \leq j \leq 2p \right\} \text{ (KTD-V)} \\ \left\{ \hat{r}_{i|i-1}^{(j)} = \hat{Q}_{\hat{\theta}_{i|i-1}^{(j)}}(s_i, a_i) - \gamma \hat{Q}_{\hat{\theta}_{i|i-1}^{(j)}}(s_{i+1}, a_{i+1}), \quad 0 \leq j \leq 2p \right\} \text{ (KTD-SARSA)} \\ \left\{ \hat{r}_{i|i-1}^{(j)} = \hat{Q}_{\hat{\theta}_{i|i-1}^{(j)}}(s_i, a_i) - \gamma \max_b \hat{Q}_{\hat{\theta}_{i|i-1}^{(j)}}(s_{i+1}, b), \quad 0 \leq j \leq 2p \right\} \text{ (KTD-Q)} \end{cases} ;$$

Compute statistics of interest;

$$\hat{r}_{i|i-1} = \sum_{j=0}^{2p} w_j \hat{r}_{i|i-1}^{(j)};$$

$$P_{\theta r_i} = \sum_{j=0}^{2p} w_j (\hat{\theta}_{i|i-1}^{(j)} - \hat{\theta}_{i|i-1}) (\hat{r}_{i|i-1}^{(j)} - \hat{r}_{i|i-1});$$

$$P_{r_i} = \sum_{j=0}^{2p} w_j \left(\hat{r}_{i|i-1}^{(j)} - \hat{r}_{i|i-1} \right)^2 + P_{n_i};$$

Correction step;

$$K_i = P_{\theta r_i} P_{r_i}^{-1};$$

$$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i (r_i - \hat{r}_{i|i-1});$$

$$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T;$$

order approaches, such as residual algorithms [9], the cost function minimized by KTD is thus biased. For the value function evaluation (extension to other cases is straightforward), the bias is:

$$\|K_i\|^2 E[\text{cov}(r_i + \gamma V_{\theta}(s') | r_{1:i-1})] \quad (9)$$

where K_i is the Kalman gain, the covariance depends on transition probabilities and the expectation is over parameters conditioned on past observed rewards. Proof, although not tricky, is not given here due to a lack of space. This bias, which is zero for deterministic transitions, is similar to the one arising from the minimization of a square Bellman residual. It favors smooth value functions [10] and acts as a regularization effect, but cannot be controlled.

3.3 Convergence Analysis

Theorem 1. Assume that posterior and noise distributions are Gaussian, and that the prior is flat (uniform distribution). Then:

$$\hat{\theta}_{i|i} = \operatorname{argmin}_{\theta} \sum_{j=1}^i \frac{1}{P_{n_j}} (r_j - g_{t_j}(\theta))^2 \quad (10)$$

Proof. KTD is a special form of a Sigma-Point Kalman Filter (SPKF) with a random walk evolution model. It is shown in [7, Ch. 4.5] that under the hypothesis of Gaussian posterior and noises, such a filter produces a *maximum a posteriori* (MAP) estimator. Thus, for KTD, $\hat{\theta}_{i|i} = \hat{\theta}_i^{\text{MAP}} = \operatorname{argmax}_{\theta} p(\theta|r_{1:i})$. Using the Bayes rule, the posterior can be rewritten as the normalized product of likelihood and prior: $p(\theta|r_{1:i}) = \frac{p(r_{1:i}|\theta)p(\theta)}{p(r_{1:i})}$. The prior is assumed flat, and the denominator does not depend on parameters, so MAP resumes to maximum likelihood. Moreover, the noise being white, the joint likelihood is the product of local likelihoods: $\hat{\theta}_{i|i} = \operatorname{argmax}_{\theta} p(r_{1:i}|\theta) = \operatorname{argmax}_{\theta} \prod_{j=1}^i p(r_j|\theta)$. As the noise is assumed Gaussian, $r_j|\theta \sim \mathcal{N}(g_{t_j}(\theta), P_{n_j})$, and maximizing a product of likelihood is equivalent to minimizing the sum of their negative logarithms: $\hat{\theta}_{i|i} = -\operatorname{argmin}_{\theta} \sum_{j=1}^i \ln(p(r_j|\theta)) = \operatorname{argmin}_{\theta} \sum_{j=1}^i \frac{1}{P_{n_j}} (r_j - g_{t_j}(\theta))^2$. \square

The form of the minimized cost function strengthen the parallel drawn in Sec. 3.2 between KTD and square Bellman residual minimization. It can also be shown (see again [7, Ch. 4.5.1]) that a SPKF (and thus KTD) update is actually an online form of a modified Gauss-Newton method, which is a variant of natural gradient descent. In this case, the Fisher information matrix is $P_{i|i}^{-1}$. Natural gradient approach has been shown to be quite efficient for direct policy search [11] and actor-critic [12], so it lets envision good empirical results for KTD. This may be considered as the first RL value (and Q -) function approximation algorithm (in a pure critic sense) involving natural gradient.

4 Experiments

4.1 Boyan Chain

The first experiment is the Boyan chain [13]. The aim is to illustrate the bias caused by stochastic transitions and to show sample-efficiency and tracking ability of KTD-V on a deterministic version of this experiment.

Stochastic Case. The Boyan chain is a 13-state Markov chain where state s^0 is an absorbing state, s^1 transits to s^0 with probability 1 and a reward of -2, and s^i transits to either s^{i-1} or s^{i-2} , $2 \leq i \leq 12$, each with probability 0.5 and reward -3. In this experiment, KTD-V is compared to TD [1] and LSTD [8]. The feature vectors $\phi(s)$ for states s^{12} , s^8 , s^4 and s^0 are respectively $[1, 0, 0, 0]^T$, $[0, 1, 0, 0]^T$, $[0, 0, 1, 0]^T$ and $[0, 0, 0, 1]^T$. The feature vectors for other states are obtained by linear interpolation. The approximated value function is thus $\hat{V}_{\theta}(s) = \theta^T \phi(s)$. The optimal value function is linear in these features, and $\theta^* = [-24, -16, -8, 0]^T$. The error measure is $\|\theta - \theta^*\|$.

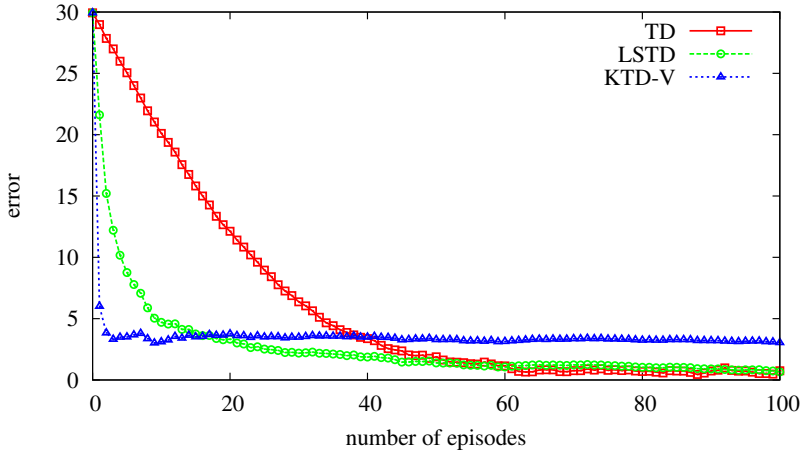


Fig. 1. Boyan Chain: stochastic case

The discount factor γ is set to 1 in this episodic task. For TD, the learning rate is set to $\alpha = 0.1$. For LSTD, the prior is set to $P_{0|0} = I$ where I is the identity matrix. For KTD-V, the same prior is used, the observation noise is set to $P_{n_i} = 10^{-3}$ and the process noise covariance to $P_{v_i} = 0I$. Choosing these parameters requires some practice, but no more than choosing a learning rate for other algorithms. Fig. 1 shows results. LSTD converges faster than TD, as expected, and KTD-V converges even faster than LSTD. However it does not converge to the optimal parameter vector, which is explained by the fact that the minimized cost-function is biased.

Deterministic and Non-Stationary Case. The Boyan chain is made deterministic by setting the probability of transiting from s^i to s^{i-1} to 1. KTD-V is again compared to LSTD and TD. Moreover, to simulate non-stationarity, the sign of the reward is switched from the 100th episode. The optimal value function is still linear in the feature vectors, and optimal parameters are $\theta_{(-)}^* = [-35, -23, -11, 0]^T$ before the MDP change, and $\theta_{(+)}^* = -\theta_{(-)}^*$ after. Algorithms parameters are the same, except the process noise covariance which is set to $P_{v_i} = 10^{-3}I$. Results are presented in Fig. 2. Here again KTD-V converges much faster than LSTD and TD, however now to the correct optimal parameter vector. After the change in reward, LSTD is very slow to converge, because of the induced non-stationarity. TD can track the correct parameter vector faster, the learning rate being constant. However, KTD converges again faster. Thus, KTD-V fails to handle the stochastic case as expected, however it converges much faster than LSTD or TD in the deterministic one. Moreover, it handles well non-stationarity, as desired.

4.2 Mountain Car

The last experiment is the mountain car task (see [11] for a full description). The objective here is to illustrate behavior of algorithms in an optimistic policy iteration scheme:

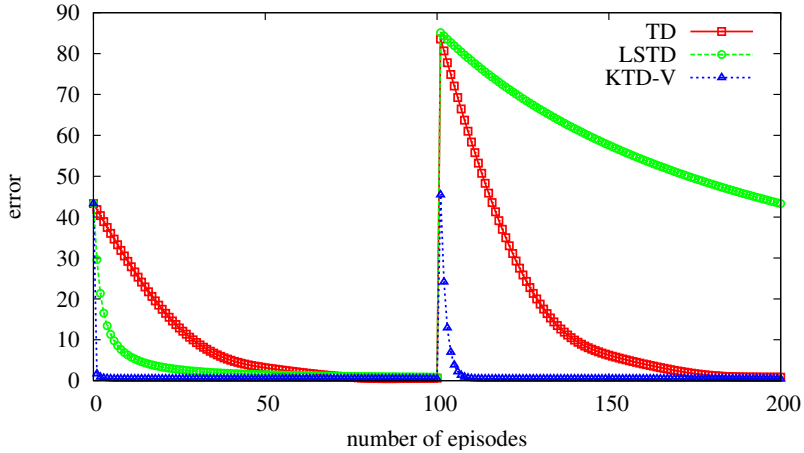


Fig. 2. Boyan Chain: deterministic and non-stationary case

learning while controlling induces non-stationary value dynamics. This task consists in driving an underpowered car up a steep mountain road, the gravity being stronger than the car engine. The discount factor is set to 0.95. State is normalized, and the parameterization is composed of a constant term and a set of 9 equispaced Gaussian kernels (centered in $\{0, 0.5, 1\} \times \{0, 0.5, 1\}$ and with a standard deviation of 0.5) for each action. This experiment compares SARSA with Q -function approximation, LSTD and KTD-SARSA within an optimistic policy iteration scheme. The followed policy is ε -greedy, with $\varepsilon = 0.1$. For SARSA, the learning rate is set to $\alpha = 0.1$. For LSTD the prior is set to $P_{0|0} = 10I$. For KTD-SARSA, the same prior is used, and the noise variances are set to $P_{n_i} = 1$ and $P_{v_i} = 0.05I$. For all algorithms the initial parameter vector is set to zero. Each episode starts in a random position and velocity uniformly sampled from the given bounds. A maximum of 1500 steps per episode is allowed. For each trial, learning is done for 200 episodes, and Fig. 3 shows the length of each learning episode averaged over 300 trials. KTD-SARSA performs better than LSTD, which performs better than SARSA with Q -function approximation. Better results have perhaps been reported for SARSA with tile-coding parameterization in the literature, however the chosen parameterization is rather crude and involves much less parameters. Moreover, even with tile-coding and optimized parameters, SARSA with function approximation is reported to take about 100 episodes to reach an optimal policy [11 Ch. 8.4.], which is about an order of magnitude higher than KTD. The optimistic policy iteration scheme used in this experiment implies non-stationarity for the learned Q -function, which explains that LSTD fails to learn a near-optimal policy. This is confirmed by [2]. LSTD has been extended to LSPI [14], which allows searching an optimal control more efficiently, however it is a batch algorithm which does not imply to learn while controlling, so it is not considered here. KTD-SARSA performs well, and learns a near-optimal policy after only a few tens of steps. Learning is also more stable with it.

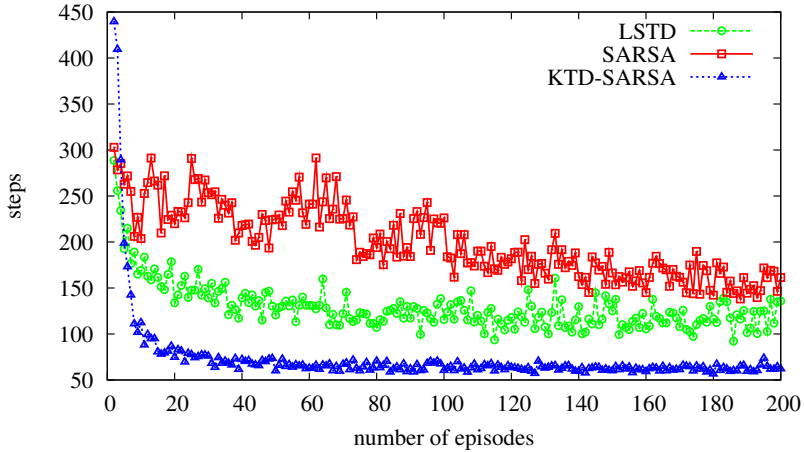


Fig. 3. Mountain car

5 Conclusion

In this paper we proposed the use of a stochastic framework (the Kalman Temporal Differences framework) for value function approximation to handle non-stationarity in RL. Its ability to handle non-stationarity has been shown experimentally (both for non-stationary system and for the case of interlaced learning and control) as well as its sample efficiency. KTD minimizes a square Bellman residual in a natural-gradient descent-like scheme which links it to other promising approaches. Computational (and memory) cost is quadratic. The KTD framework compares favorably to state-of-the-art algorithms, however there is still room for improvement. First, the resulting estimator is biased for stochastic transitions. It would be a great advantage to handle stochastic MDP, and [10,15,16] are interesting leads. When using KTD, the practitioner has to choose a prior, a process noise and an observation noise, which are domain-dependent; there exists a vast literature on adaptive filtering for traditional Kalman filtering, and adaptation to KTD is possible. Using a KTD-based function evaluation in an actor-critic architecture can also be envisioned. For example, incremental natural actor-critic algorithms are presented in [17]. TD is used as the actor part instead of LSTD, mostly because of the inability of the latter one to handle non-stationarity. In this case, we argue that KTD is an interesting alternative for the critic part. Finally, as this framework can handle nonlinear parameterization, it can be of interest to combine it with neural networks or basis adaptation schemes. A kernel-based nonlinear parameterization is used in [18].

Acknowledgements

Olivier Pietquin thanks the European Community (FP7/2007-2013, grant agreement 216594, CLASSiC project : www.classic-project.org) and the Région Lorraine for financial support.

References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1996)
2. Phua, C.W., Fitch, R.: Tracking Value Function Dynamics to Improve Reinforcement Learning with Piecewise Linear Function Approximation. In: International Conference on Machine Learning, ICML 2007 (2007)
3. Sutton, R.S., Koop, A., Silver, D.: On the role of tracking in stationary environments. In: Proceedings of the 24th international conference on Machine learning, pp. 871–878 (2007)
4. Geist, M., Pietquin, O., Fricout, G.: Kalman Temporal Differences: the deterministic case. In: Proceedings of the IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL 2009), Nashville, TN, USA (April 2009)
5. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. Transactions of the ASME—Journal of Basic Engineering 82(Series D), 35–45 (1960)
6. Julier, S.J., Uhlmann, J.K.: Unscented filtering and nonlinear estimation. Proceedings of the IEEE 92(3), 401–422 (2004)
7. van der Merwe, R.: Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models. PhD thesis, Oregon Health&Science University, Portland, USA (2004)
8. Bradtke, S.J., Barto, A.G.: Linear Least-Squares Algorithms for Temporal Difference Learning. Machine Learning 22(1-3), 33–57 (1996)
9. Baird, L.C.: Residual Algorithms: Reinforcement Learning with Function Approximation. In: Proceedings of the International Conference on Machine Learning, pp. 30–37 (1995)
10. Antos, A., Szepesvári, C., Munos, R.: Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. Machine Learning 71(1), 89–129 (2008)
11. Kakade, S.: A natural policy gradient. In: Advances in Neural Information Processing Systems 14 (NIPS 2001), Vancouver, British Columbia, Canada, pp. 1531–1538 (2001)
12. Peters, J., Vijayakumar, S., Schaal, S.: Natural actor-critic. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 280–291. Springer, Heidelberg (2005)
13. Boyan, J.A.: Technical Update: Least-Squares Temporal Difference Learning. Machine Learning 49(2-3), 233–246 (1999)
14. Lagoudakis, M.G., Parr, R.: Least-Squares Policy Iteration. Journal of Machine Learning Research 4, 1107–1149 (2003)
15. Jo, S., Kim, S.W.: Consistent Normalized Least Mean Square Filtering with Noisy Data Matrix. IEEE Transactions on Signal Processing 53(6), 2112–2123 (2005)
16. Engel, Y., Mannor, S., Meir, R.: Reinforcement Learning with Gaussian Processes. In: Proceedings of International Conference on Machine Learning, ICML 2005 (2005)
17. Bhatnagar, S., Sutton, R.S., Ghavamzadeh, M., Lee, M.: Incremental Natural Actor-Critic Algorithms. In: Advances in Neural Information Processing Systems, Vancouver, vol. 21 (2008)
18. Geist, M., Pietquin, O., Fricout, G.: Bayesian Reward Filtering. In: Girgin, S., Loth, M., Munos, R., Preux, P., Ryabko, D. (eds.) EWRL 2008. LNCS (LNAI), vol. 5323, pp. 96–109. Springer, Heidelberg (2008)

Ensembling Heterogeneous Learning Models with Boosting

Diego S.C. Nascimento and André L.V. Coelho

Graduate Program in Applied Informatics, University of Fortaleza
Av. Washington Soares 1321/J30, Fortaleza-CE, Brazil, 60811-905
silveiraal@gmail.com, acoelho@unifor.br

Abstract. In this paper, we investigate the potentials of a novel classifier ensemble scheme, referred to as *heterogeneous boosting* (HB), which aims at delivering higher levels of diversity by allowing that distinct learning algorithms be recruited to induce the different components of the boosting sequence. For the automatic design of the HB structures in accord with the nuances of the problem at hand, a genetic algorithm engine is adopted to work jointly with AdaBoost, the state-of-the-art boosting algorithm. To validate the novel approach, experiments involving well-known learning algorithms and classification datasets from the UCI repository are discussed. The accuracy, generalization, and diversity levels incurred with HB are matched against those delivered by AdaBoost working solely with RBF neural networks, with the first either significantly prevailing over or going in par with the latter in all the cases.

Keywords: Boosting, heterogeneous models, genetic algorithms.

1 Introduction

Over the last two decades, numerous theoretical and experimental studies in Machine Learning (ML) have supported the idea that pooling the decisions of different estimators within a unique combination structure can lead to substantial improvements both in terms of training accuracy and learning generalization. As a consequence, several approaches for designing ensembles of estimators have been conceived so far [12], each trying to properly induce and exploit the local different behaviors of the base predictors in order to enhance the overall system performance. Among these approaches, two classes have received increasing attention, namely, those using different subsets/configurations of training data jointly with a single learning method and those adopting different learning methods associated with different predictors.

As the key for the success of any ensemble lies in how its components disagree on their predictions [12], the methods of the second class try to foster high levels of diversity by making use of heterogeneous (also called hybrid) architectures comprising different types of learning algorithms [3,4,10]. By this means, the learned components are produced by different search processes taking place over

dissimilar configurations of the hypothesis space. So, it is expected that they correspond to different perspectives over the same training data and are thus endowed with disparate levels of expertise and accuracy [5]. On the other hand, methods of the first class try to promote ensemble diversity by altering aspects of the training set over which each classifier is generated. This is usually achieved via data resampling, rotation, distortion, the use of different data sources, and/or adoption of different pre-processing schemes [9].

Boosting [7] and bagging [2] allude to two representative families of data resampling ensemble methods. While the latter generates multiple bootstrapped sets over the training data and aggregates the outputs of the resulting estimators via a simple majority vote (MV) rule, the former operates iteratively so that the data distribution in a round is changed adaptively to be overrepresented by samples mispredicted in the previous round. By this means, boosting entails a sort of hierarchical process of ensemble creation, where models produced at later stages of the sequence learn to discriminate more complex regions of the input space. It is theoretically shown that a combination of a sequence of moderately inaccurate estimators can deliver an error rate that is arbitrarily small on the training data, and for this purpose boosting uses the performance indices exhibited by these “weak” models as their weights for voting. Regardless of their conceptual differences, the standard settings of boosting and bagging, as well as of some methodologies hybridizing them [9,13], stipulate that the same learning algorithm is used for generating all ensemble components.

Even though the two classes of ensemble creation approaches discussed above are complementary in their operational basis, their combination into a unique conceptual framework has not been deeply investigated so far. To fill this gap, in this paper, we introduce a novel boosting scheme, referred to as *heterogeneous boosting* (HB), which aims at delivering higher levels of ensemble diversity by allowing that different learning algorithms be recruited to induce the weak components over the resampled data. As different blueprints of heterogeneous ensembles may yield distinct results in terms of performance (due to the sequential nature of boosting), achieving the best ensemble design for a given problem turns out to be a very difficult combinatorial optimization problem. For tackling it, a genetic algorithm (GA) engine is adopted in HB to work jointly with AdaBoost [7,12], the most popular realization of boosting. As a manner to validate the novel approach, experiments involving 10 well-known learning algorithms and 18 well-known pattern classification datasets taken from the UCI repository are discussed here. The accuracy, generalization, and diversity levels achieved by HB are contrasted with those produced by standard AdaBoost operating solely with RBF neural networks (NNs) [8], since combinations of neural networks are typical ensemble settings investigated in the literature [11,15].

In the following section, we describe aspects related to boosting and then provide details on HB and its design via a customized GA. In Section 3, we present and discuss the results achieved in the experiments conducted. Section 4 concludes the paper, bringing remarks on future work.

2 Heterogeneous Boosting via a Genetic Algorithm

In a nutshell, boosting is based on the observation that finding many simple estimation rules can be much easier than finding a single, highly accurate predictor [7]. In the terminology adopted, these simple rules, whose error rates are only slightly better than random guessing, are said to be produced by a weak learner. In contrast, a strong learner is that capable of generating models that are arbitrarily well correlated with the true data estimation. The purpose of boosting is thus to sequentially apply a weak learning algorithm to repeatedly modified versions of the data, thereby producing a series of weak estimators. The predictions from all weak models are then combined via a weighted MV rule (with the weights being proportional to each classifier's accuracy on its associated training set) towards the development of a stronger model [12].

The data modifications at each boosting step consist of applying weights to each training sample. According to Kotsiantis & Pintelas [9], there are two ways that AdaBoost can make use of these weights to construct a new training set to give to the weak learner. In boosting by sampling, examples are drawn with replacement with probability proportional to their weights. So, the derived training sets all have the same size of the original dataset, but the examples within them are chosen stochastically, like in bagging. Conversely, in boosting by weighting, although the derived training sets are identical to the original one, they are augmented with weights measuring the hardness in classifying each sample alone. This latter approach, which is followed by AdaBoost.M1, the standard version of AdaBoost for classification, has the clear advantage that each example is incorporated (at least in part) in the training set [11].

Boosting shows some resemblance in structure to bagging. However, unlike bagging, which is largely a variance reduction method [2], boosting appears to reduce both bias and variance [9,13]. This is because boosting attempts to correct the bias of the most recently constructed model by forcing it to concentrate its efforts on instances that have not been correctly learned. A number of experimental studies comparing boosting algorithms (Arcing and AdaBoost) and bagging suggest that they have quite different behaviors.

In [11], they are contrasted in terms of several criteria on several well-known datasets (most of which taken from the UCI repository [1]) using feedforward NNs or decision trees as base learners. Among several conclusions, the authors point out that: 1) while bagging is almost always more accurate than a single classifier, it is sometimes much less accurate than boosting; 2) bagging is usually more consistent than boosting, which can create ensembles that are less accurate than a single classifier and seems to be affected by the characteristics of the dataset being examined; and 3) boosting ensembles may overfit noisy data.

With the purpose of increasing the diversity levels of the ensemble models produced by boosting, we have conceived the idea of adopting different learning algorithms for possibly inducing the sequence of ensemble components over the resampled data, giving birth to the HB scheme. Indeed, recent work [3,4] investigating the impact of varying the number and type of ensemble members on the performance of some combination methods has empirically shown that

hybrid structures usually behave significantly better in terms of accuracy and diversity than non-hybrid ones. Moreover, Menahem et al. [10] have evaluated the impact of the choice of combining methods on multi-inducer ensembles created specifically for copying with malware detection. In those studies, however, the ensemble components produced by the different learners were all induced over the same training data, as no resampling or other data-varying methods were effectively employed.

In its current version, heterogeneous boosting adopts 10 state-of-the-art learning algorithms representing five distinct classes of classifier inducers to work within AdaBoost.M1. These algorithms are available in Weka, a well-known ML toolkit [14]. They comprise: i) Simple Naïve Bayes, founded on Bayesian statistics; ii) RBF NNs and Support Vector Machines (SVMs) trained with SMO algorithm, both based on non-linear function representations; iii) J48, Decision stump, and REP Tree, working with decision trees; iv) IBk, an instance learning algorithm; and v) OneR, PART and Decision table, which generate hypothesis in rule format. The choice of these learning algorithms is due to the different representation and searching bias they incur in their functioning, possibly yielding weak estimators with complementary roles.

One important aspect to be taken into account is that the application of distinct sequential orders of heterogeneous models via boosting may entail very different results in terms of ensemble performance. This is because different sequences may correspond to distinct problem decompositions and the performance of a model induced in a certain boosting round depends very much on the weighted data received for training. So, it is very reasonable to accept that, for a given round, a particular type of inducer may be more appropriate than others to be applied.

We have modeled the task of specifying the best heterogeneous structure in accordance with the nuances of the prediction problem in sight as a combinatorial optimization problem. As the size of the search space is of an exponential nature, namely $O(K^M)$, where M denotes the number of different inducers available and K the number of ensemble components to be induced, handling this task via conventional optimization methods turns out to be computationally intractable, even for moderate magnitudes of M and K . Therefore, in HB, a customized GA engine has been deployed for such a purpose. As a typical class of evolutionary algorithms, GAs comprehend a family of stochastic search and optimization algorithms inspired from the mechanics of Natural Selection and concepts of population genetics [6]. According to the GA framework, candidate solutions to a given problem play the role of individuals in a population, while a fitness function determines the environment within which the solutions “live” and have their levels of adaptation (quality) measured. Here, optimal solutions emerge through the evolution of the population, which takes place after the repeated application of some operators mimicking well-known natural phenomena: parent (mating) selection, recombination, mutation, and survivor (environmental) selection.

In the evolutionary engine of HB, each individual of the population (which is initially randomly generated) represents a whole ensemble structure and is

codified as a K -size linear array of integer values. For the k -th position (gene), $M + 1$ values (alleles) are available to be selected, one for each type of inducer in the repertory and another indicating the possibility of component pruning. Usually, the pruning of ensemble components happens as a second stage in ensemble creation (after the generation of components), aiming at increasing accuracy by reducing the redundancy and complexity of the resulting ensemble model [15]. In our case, component generation and pruning occurs simultaneously, allowing HB to tune the ensemble size in agreement with the problem's demands.

As fitness function, a convex linear combination of two terms have been adopted: one related to accuracy and another to parsimony. While the first term refers to the cross-validation error delivered in training (see Section 3), the second captures the complexity of the ensemble model associated to an individual. By this means, the lower the cross-validation error and the number of components of an ensemble model, the higher will be the fitness of its affiliated GA individual. Moreover, the Roulette Wheel operator [6] is used both for selecting individuals to reproduce (among parents) and to survive to the next generation (among parents and offspring), even though elitism (salvation of the best current individual) is also adopted in this last phase. Individuals are recombined through a single-point crossover and the resulting offspring undergo modifications via creep mutation. The stop criterion adopted is to go through a given number of generations of evolution.

3 Empirical Assessment

To assess the performance of HB, a prototype was developed under Weka [14] and extensive experiments have been conducted over several UCI benchmark datasets [1]. These datasets are indicated in Table 1 and their description in terms of type of attribute values and number of instances, attributes, and classes can be found elsewhere [9,11]. To serve as yardstick against which we could match the performance of HB ensemble models, we have also recorded results delivered by AdaBoost.M1 working only with RBF NNs. This choice holds for combinations of neural models are typical ensemble settings investigated in the literature [11,15] and because these models are known to show high error variance [8]. In particular, an advantage of RBF NNs over multilayer perceptrons (another popular NN model) is that the training of each layer of neurons in these networks can be conducted separately, thus yielding efficiency [8].

Aiming at delivering statistically significant results, for each dataset, 10 pairs of training/test (66,6%/33,4%) partitions were randomly generated in a stratified manner (i.e. with preservation of class distributions) by using 10 different random seeds. Over the training partitions, both homogeneous/heterogeneous boosting settings were executed under the frame of a 10-fold stratified cross-validation process [14]. In particular, the error rates produced in this manner served as scores to guide the GA engine (see Section 2). Conversely, test data were used for assessing the levels of generalization achieved by the resulting ensemble models trained ultimately over the whole training partition.

Table 1. Performance comparison between homogeneous and heterogeneous ensemble models

Dataset	Standard AdaBoost.M1 with RBF NNs				Heterogeneous Boosting				t -test
	Training	Test	Q -Stat.	Training	Test	Q -Stat.			
anneal	0.0477±0.0099	0.0443±0.0128	0.0333	0.0165±0.0042	0.0262±0.0111	-0.0309	0.01		
breast-cancer	0.3173±0.0584	0.3330±0.0289	0.0748	0.1245±0.0243	0.3160±0.0347	0.1205	0.17		
bupa	0.3983±0.0607	0.3727±0.0427	0.1649	0.2076±0.0249	0.3489±0.0752	0.0150	0.40		
colic	0.2262±0.0460	0.2314±0.0194	-0.0044	0.1286±0.0316	0.1727±0.0249	0.2712	0.00		
credit-a	0.1872±0.0242	0.1890±0.0235	0.2123	0.1069±0.0170	0.1653±0.0153	0.0346	0.02		
diabetes	0.2672±0.0235	0.2735±0.0197	0.2046	0.2147±0.0197	0.2559±0.0239	0.1773	0.08		
glass	0.3808±0.0622	0.3560±0.0382	0.1888	0.2047±0.0234	0.3489±0.0432	0.2068	0.56		
haberman	0.2962±0.0226	0.2905±0.0268	0.2276	0.1205±0.0095	0.2726±0.0246	0.2231	0.11		
heart-c	0.1990±0.0342	0.1905±0.0259	0.0648	0.1030±0.0182	0.1889±0.0322	0.2862	0.78		
hepatitis	0.1604±0.0513	0.1627±0.0340	-0.3892	0.0917±0.0280	0.1637±0.0217	-0.1408	0.94		
ionosphere	0.0892±0.0125	0.0792±0.0138	-0.0365	0.0617±0.0185	0.0879±0.0220	-0.2867	0.30		
iris	0.0569±0.0299	0.0505±0.0117	0.5033	0.0167±0.0104	0.0505±0.0151	0.1563	1.00		
segment	0.0814±0.0078	0.0802±0.0129	0.5580	0.0000±0.0000	0.0426±0.0046	-0.0766	0.00		
sick	0.0429±0.0047	0.0430±0.0033	0.5013	0.0000±0.0000	0.0182±0.0014	-0.1622	0.00		
sonar	0.2465±0.0480	0.2190±0.0298	-0.2254	0.0000±0.0000	0.2423±0.0354	0.2239	0.05		
vehicle	0.3184±0.0328	0.3211±0.0176	0.5836	0.2142±0.0129	0.2971±0.0245	-0.0927	0.03		
vote	0.0595±0.0177	0.0523±0.0106	-0.0036	0.0311±0.0107	0.0467±0.0104	-0.2637	0.22		
zoo	0.1314±0.0335	0.1061±0.0378	1.0000	0.0400±0.0113	0.0712±0.0258	0.6747	0.01		

For the experiments, the GA control parameters were set as follows (after manual calibration): 20 as population size; 80% and 10% as crossover and mutation rates, respectively; and 20 as maximum number of generations. It is worth mentioning that the performance results reported for each dataset relate to the best weight combinations achieved for the two terms employed in the GA fitness function. Moreover, we have made extensively use of the validation testbench and code implementations (with default control parameter values) available in Weka for the 10 learning algorithms currently adopted in HB (see Section 2). Since it has been observed that for boosting most of the gain in performance is due to the first few estimators combined [9,11], we have adopted $K = 10$.

Table 1 contrasts the performance achieved with homogeneous and heterogeneous ensemble models produced by boosting in terms of accuracy (average cross-validation error), generalization (average test error), and diversity. To estimate the diversity levels among the ensemble components, we have adopted Yule's Q -statistic, calculated pairwise as [12]: $Q_{i,j} = \frac{ad-bc}{ad+bc}$, where a (d) is the fraction of samples correctly (incorrectly) classified by both classifiers i and j and b is the fraction of samples correctly classified by i but incorrectly classified by j (c is the opposite). Q assumes positive values if there is high correlations in the classifiers' outputs; and negative values, otherwise. Maximum diversity is obtained for $Q = 0$ and the final value of this statistic is averaged over all pairs of classifiers. In the table, better values for all criteria are highlighted. The last column shows the p -values delivered by paired t -test [14] with 5% significance when applied to the error values achieved by the contestants over the 10 test partitions generated for each dataset.

Overall, the results suggest that adopting HB models designed by evolution may entail apparent gains in performance. This is particularly noticeable in terms of training accuracy (they have prevailed in all datasets considered) and diversity (they have outperformed in 66,6% of the cases). Regarding learning generalization, the heterogeneous ensemble models have delivered lower average test error rates in 14 out of 18 problems. In addition, the application of t -test indicates that heterogeneous ensembles have performed statistically better in seven cases ($p < 0.05$) and have been comparable to homogeneous models in the remaining problems. As a general rule, one could conclude that embedding more heterogeneity in the weak models produced by boosting can be instrumental for achieving improvements in terms of performance. The empirical results we have obtained also support those achieved in related work on heterogeneous ensembles [3,4,10], mainly in ratifying that the choice of the types of components to be induced into an ensemble is an important issue to be properly pursued.

4 Final Remarks

In this paper, heterogeneous ensemble models produced via boosting and a customized genetic algorithm have been characterized and empirically evaluated, taking as reference the performance levels achieved by standard boosting configured with only RBF neural networks. The results confirm that sensible gains

in terms of accuracy, generalization, and diversity may be incurred by resorting to this novel approach.

As future work, we plan to conduct a more comprehensive empirical analysis involving other homogeneous ensemble models produced by standard boosting (in particular, those generated with the other types of inducers considered in this study). As well, a statistical account of the types of inducers more frequently recruited by the GA to generate the heterogeneous ensemble models is underway. Experiments with other datasets, particularly those with noisy data [9,11] are also under consideration. Finally, bringing the idea of heterogeneous models to other data resample based ensemble techniques, like bagging [2] and derived methods [9,13], shall be also investigated.

Acknowledgment

The authors gratefully acknowledge Funcap for a master degree scholarship.

References

1. Asunción, A., Newman, D.J.: UCI Machine Learning Repository. University of California at Irvine (2007), <http://ics.uci.edu/~mllearn/MLRepository.html>
2. Breiman, L.: Bagging predictors. *Mach. Learn.* 24(2), 123–140 (1996)
3. Canuto, A., Abreu, M., Oliveira, L., Xavier Jr., J., Santos, A.: Investigating the influence of the choice of the ensemble members in accuracy and diversity of selection-based and fusion-based methods for ensembles. *Pattern Recognit. Lett.* 28(4), 472–486 (2007)
4. Canuto, A., Oliveira, L., Xavier Jr., J., Santos, A., Abreu, M.: Performance and diversity evaluation in hybrid and non-hybrid structures of ensembles. In: *Procs. of the Fifth Int. Conf. on Hybrid Intelligent Systems*, pp. 285–290 (2005)
5. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) *MCS 2000. LNCS*, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
6. Eiben, A., Smith, J.: *Introduction to Evolutionary Computing*. Springer, Heidelberg (2003)
7. Freund, Y., Schapire, R.: A decision-theoretic generalization of online learning and an application to boosting. *J. Comput. Syst. Sci.* 55, 119–139 (1997)
8. Haykin, S.: *Neural Networks—A Comprehensive Foundation*. Prentice-Hall, Englewood Cliffs (1999)
9. Kotsiantis, S.B., Pintelas, P.E.: Combining bagging and boosting. *Int. J. Comput. Intell.* 1(4), 324–333 (2004)
10. Menahem, E., Shabtai, A., Rokach, L., Elovici, Y.: Improving malware detection by applying multi-inducer ensemble. *Comput. Stat. Data Anal.* 53, 1483–1494 (2009)
11. Opitz, D., Maclin, R.: Popular ensemble methods: An empirical study. *J. Artif. Intell. Res.* 11, 169–198 (1999)
12. Polikar, R.: Ensemble based systems in decision making. *IEEE Circuits Syst. Mag.* 6(3), 21–45 (2006)
13. Webb, G.I.: Multiboosting: A technique combining boosting and wagging. *Mach. Learn.* 40, 159–196 (2000)
14. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, Amsterdam (2005)
15. Zhou, Z.-H., Wu, J., Tang, W.: Ensembling neural networks: Many could be better than all. *Artif. Intell.* 137(1-2), 239–263 (2002)

Improvement Algorithm for Approximate Incremental Learning

Tadahiro Oyama¹, H. Kipsang Choge¹, Stephen Karungaru¹, Satoru Tsuge¹,
Yasue Mitsukura², and Minoru Fukumi¹

¹ University of Tokushima, 2-1, Minami-Josanjima, Tokushima, 770-8506, Japan
{oyama,choge,karunga,tsuge,fukumi}@is.tokushima-u.ac.jp

² Tokyo University of Agriculture and Technology, 2-24-16, Naka-cho,
Higashi-koganei, Tokyo, 184-8588, Japan
mitsu_e@cc.tuat.ac.jp

Abstract. This paper presents an improved algorithm of Incremental Simple-PCA. The Incremental Simple-PCA is a fast incremental learning algorithm based on Simple-PCA. This algorithm need not hold all training samples because it enables update of an eigenvector according to incremental samples. Moreover, this algorithm has an advantage that it can calculate the eigenvector at high-speed because matrix calculation is not needed. However, it had a problem in convergence performance of the eigenvector. Thus, in this paper, we try the improvement of this algorithm from the aspect of convergence performance. We performed computer simulations using UCI datasets to verify the effectiveness of the proposed algorithm. As a result, its availability was confirmed from the standpoint of recognition accuracy and convergence performance of the eigenvector compared with the Incremental Simple-PCA.

Keywords: PCA, Simple-PCACincremental learningCdimensional reductionCpattern recognition.

1 Introduction

In recent years, the high-dimensional data with the various features can be easily obtained by the increase in capacity of a storage medium. An opportunity to treat the high-dimensional data has been increasing in the fields such as machine learning and data mining. However, a phenomenon that causes significant error appears when the dimension of data becomes too high. This is called “curse of dimensionality” [1]. The dimensional reduction is effective as the technique to solve this problem. There is the principal component analysis (PCA) as a typical technique of this dimensional reduction. Originally, it has been used as a technique to derive the low feature variables from the multivariate data in the area of the multivariate analysis etc. However, it is often used as a method of the dimensional reduction and the feature extraction in the field of the pattern recognition. Its effectiveness has been shown in the areas such as face recognition [2], industrial robotics [3], and 3-D object recognition [4].

When the PCA is applied to various systems in the real world, we are confronted with two problems. One is that a complete training dataset cannot be obtained beforehand. In the real world, increase of training and evaluation data is expected. It is thought that the accuracy can be improved more by utilizing these data that increased. Therefore, the sequential learning with increase of data is necessary. As an algorithm which achieves this, a technique called Incremental PCA (IPCA) that equipped PCA with incremental learning function exists. IPCA was an algorithm proposed by P.M.Hall et al. [5], and it has been used for the localization control of a mobile robot [6][7] and online image processing [8].

The other problem is the necessity of the matrix calculation when PCA is performed. Since the PCA needs to solve an eigenvalue problem, the amount of calculation increases exponentially. Simple-PCA was proposed as the technique to solve such a problem. Simple-PCA is an approximation algorithm of PCA, and was developed by M. Partridge et al. [9]. In addition, its availability has been reported on many fronts, for example, recognition of hand-written characters [9], dimensionality reduction of a model for information retrieval [10], recognition using face images [11][12].

Since it is necessary to solve an eigenvalue problem also in above-mentioned IPCA, matrix calculation is essential. If incremental learning becomes possible in Simple-PCA algorithm that used repeated computation, high-speed incremental learning is promising. Thereby, the application range in the real world will spread. In particular, it is available in the built-into system that should operate in real time. We already proposed the technique called Incremental Simple-PCA as the sequential learning type algorithm of Simple-PCA [13]. This method updates approximately the eigenvector by using the last eigenvector and the incremental data which were obtained by the calculation of Simple-PCA. However, the problem was seen in respect of the convergency of the updated eigenvector. Thereby, in this paper, we try improvement which focused on the convergence of the eigenvector of this Incremental Simple-PCA. The validity of improved Incremental Simple-PCA is verified from the viewpoints of accuracy, a computational time and a memory usage by using dataset.

The rest of this paper is organized as follows. In Section 2, the algorithm of basic Simple-PCA and the algorithm of improved Incremental Simple-PCA which is the proposal technique are explained. Chapter 3 presents the results and the discussions in incremental learning experiments that used UCI dataset. Finally, the conclusion and future works are in Section 4.

2 Simple-PCA and Incremental Algorithm

2.1 Algorithm of Simple-PCA

Simple-PCA (Simple Principal Component Analysis) is a technique proposed by Partridge et al. [9] to speed up principal component analysis. The technique is an approximation algorithm from which principal components can be sequentially found from the first component. Its effectiveness has been confirmed in many

fields such as recognition of hand-written characters, dimensionality reduction of a model for information retrieval, recognition using face images and so on [9,10,11,12]. The algorithm of this technique sequentially solves for eigenvectors that maximizes the variance over all samples. Concretely, it is summarized as follows.

First of all, a set of vectors to use is defined as follows.

$$\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\} \tag{1}$$

To make the center of gravity of this set the origin, the calculation shown in eq. (2) is performed, and a new set of vectors (3) is obtained.

$$\mathbf{x}_i = \mathbf{v}_i - \frac{1}{m} \sum_{j=1}^m \mathbf{v}_j \tag{2}$$

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\} \tag{3}$$

Next, the following output function is used.

$$y_n = (\boldsymbol{\alpha}_n^k)^T \mathbf{x}_j \tag{4}$$

where $\boldsymbol{\alpha}_n^k$ is an eigenvector that represents the n-th principal component and k is number of repetitions. Initially, the initial vector $\boldsymbol{\alpha}_n^0$ can be set to any vector. If the input vector \mathbf{x}_j has the same direction component as $\boldsymbol{\alpha}_n^k$, the function shown by eq. (4) outputs a positive value. If it has the opposite direction component, a negative value is used. Thus, the following threshold function is introduced.

$$\mathbf{f}(y_n, \mathbf{x}_j) = \begin{cases} \mathbf{x}_j & \text{if } y_n \geq 0 \\ -\mathbf{x}_j & \text{otherwise} \end{cases} \tag{5}$$

The initial vector $\boldsymbol{\alpha}_n^0$ initialized by arbitrary random values is brought close in the same direction as $\boldsymbol{\alpha}_n$ by these functions and the repetition operation shown in eq. (6).

$$\boldsymbol{\alpha}_n^{k+1} = \frac{\sum_j \mathbf{f}(y_n, \mathbf{x}_j)}{\|\sum_j \mathbf{f}(y_n, \mathbf{x}_j)\|} \tag{6}$$

where $\boldsymbol{\alpha}_n^{k+1}$ is a vector after calculating $k + 1$ times. The value of the output function is calculated by using $\boldsymbol{\alpha}_n^k$ which is the previous calculation result. Furthermore, this repetition calculation is done until $\boldsymbol{\alpha}_n^{k+1}$ is converged. This vector obtained after it converges is an eigenvector.

When the next eigenvector is calculated, it is necessary to calculate it by using a new vector \mathbf{x}'_j after the previous principal component is removed from the input vector by doing the calculation shown in eq. (7).

$$\mathbf{x}'_j = \mathbf{x}_j - (\boldsymbol{\alpha}_n^{k+1} \cdot \mathbf{x}_j) \boldsymbol{\alpha}_n^{k+1} \tag{7}$$

After the component is removed, the principal component can be evaluated by repeating a similar calculation in order with a high accumulated relevance.

2.2 Algorithm of Improved Incremental Simple-PCA

This section explains the algorithm of improved Incremental Simple-PCA which is proposed in this paper. The previous Incremental Simple-PCA updated the eigenvector by using the incremental sample, the mean vector and the eigenvector which was finally derived. The specific algorithm is shown below.

First, the current mean vector of all samples and the new incremental sample are defined as $\bar{\mathbf{v}}$ and \mathbf{v}_{M+1} , respectively. M shows the number of all samples. The improved Incremental Simple-PCA performs update of the eigenvector α_n^k using the incremental sample \mathbf{v}_{M+1} , the mean vector $\bar{\mathbf{v}}$ and the repetition calculation result $\mathbf{f}_n^k (n = 1, \dots, L; k = 1, \dots, K_n)$ at each time in Simple-PCA. Here, n and k express the number of principal component vectors (eigenvectors) and the iteration count of calculation performed by Simple-PCA. Thus, \mathbf{f}_n^k means the computation result obtained by k -th operation at the time of calculation of n -th eigenvector by Simple-PCA. Furthermore, L is the number of eigenvectors calculated by Simple-PCA, and K_n is iterative calculation frequency needed when the n -th eigenvector is found. For this reason, it can be said that it is necessary to hold $L \times K_n$ calculation results to execute improved Incremental Simple-PCA proposed in this paper.

Actually equations are as follows. The new mean vector $\bar{\mathbf{v}}'$ is obtained by updating the mean vector of all samples $\bar{\mathbf{v}}$ using eq. (8).

$$\bar{\mathbf{v}}' = \frac{1}{M + 1} (M\bar{\mathbf{v}} + \mathbf{v}_{M+1}) \tag{8}$$

The new input sample \mathbf{x}_{M+1} is calculated using this updated mean vector $\bar{\mathbf{v}}'$.

$$\mathbf{x}_{M+1} = \mathbf{v}_{M+1} - \bar{\mathbf{v}}' \tag{9}$$

Next, the threshold function is introduced as well as the case of Simple-PCA.

$$y_n = (\alpha_n^k)^T \mathbf{x}_{M+1} \tag{10}$$

$$\mathbf{f}_n^{k'}(y_n, \mathbf{x}_{M+1}) = \begin{cases} \mathbf{x}_{M+1} & \text{if } y_n \geq 0 \\ -\mathbf{x}_{M+1} & \text{otherwise} \end{cases} \tag{11}$$

α_n^k expresses the eigenvector after the k -th calculation at the time of deriving the n -th eigenvector, and it can be obtained by normalizing \mathbf{f}_n^{k-1} . The new eigenvector $\alpha_n^{k'}$ is found by adding and normalizing $\mathbf{f}_n^{k'}$ and \mathbf{f}_n^k obtained in eq. (11).

$$\alpha_n^{k'} = \frac{\mathbf{f}_n^k + \mathbf{f}_n^{k'}}{\|\mathbf{f}_n^k + \mathbf{f}_n^{k'}\|} \tag{12}$$

The calculation result of the k -th times is obtained in order to find the n -th eigenvector by executing from eq. (10) to eq. (12). In other words, to obtain the final eigenvector, it is necessary k times to repeat the calculation of eq. (12) from eq. (10). In order to get the next eigenvector, the calculation shown in eq. (10) is

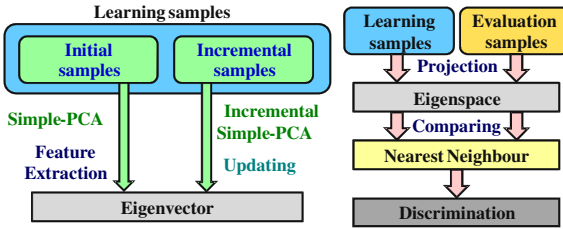


Fig. 1. The flow of experiment

performed by using the eigenvector updated in the last calculation. The eigenvector is updated by executing these k times repetition calculations according to the incremental sample. Finally, the component of the present eigenvector is removed from the incremental sample by using eq. (13) to obtain the next eigenvector.

$$\mathbf{x}'_{M+1} = \mathbf{x}_{M+1} - (\boldsymbol{\alpha}_n^{k'} \cdot \mathbf{x}_{M+1})\boldsymbol{\alpha}_n^{k'} \quad (13)$$

Afterwards, the eigenvector can be sequentially updated by executing the similar repetition calculation from eq. (8) every time the sample is added.

3 Verification Experiment

3.1 Flow of Experiment

We carry out a recognition experiment to verify the performance of proposed improved Incremental Simple-PCA. In this paper, the incremental learning experiment is conducted by using the dataset of UCI Machine Learning Repository [14]. The flow of the experiment is as shown in Fig. 1. In this experiment, we prepare learning samples and evaluation samples. The learning samples are divided into the samples for the initial learning and the incremental learning. Simple-PCA is executed using the initial samples for learning. As a result, the eigenvector in the initial state is obtained. The learning samples and the evaluation samples are projected to the eigenspace by using this eigenvector. The discrimination is conducted by the nearest neighbor method using the Euclidean distance of the evaluation samples and the learning samples. Therefore, the recognition result in the initial state can be obtained. Next, for the incremental learning, the eigenvector is updated by performing improved Incremental Simple-PCA using the incremental samples. Each time it is updated, the discrimination is carried out using the updated one.

3.2 Experimental Conditions

In this experiment, iris in UCI Machine Learning Repository [14] is used as a dataset for discrimination. The iris data has information on length and width of sepal and petal. In this dataset, the task classifying into three kinds of classes is

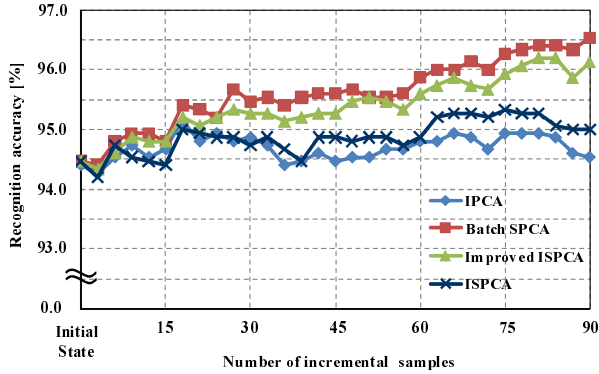


Fig. 2. Change of the recognition accuracy by incremental learning

carried out, and 50 samples per class are prepared. As a basic policy of experiment, we construct initial feature space using about 20% of the total datasets, the other about 20% datasets is used for testing. The remainder is used for incremental learning. Therefore, the number of samples for the initial learning, the incremental learning, and the evaluation are 30, 90, 30 respectively. We assume the sample chosen by each class one by one to be one set. The experiment is conducted by the number of times equal to the number of sets while exchanging the sets of the initial, the incremental, and the evaluation. We carried out the all experiments on Intel Core 2 Quad 2.4GHz CPU and 4GB RAM PC. Moreover, we experiment by using not only improved Incremental Simple-PCA but also IPCA [5] which used matrix calculation, the batch Simple-PCA and the previous Incremental Simple-PCA [13]. In this paper, batch Simple-PCA means Simple-PCA to the batch data (initial data and incremental data).

3.3 Recognition Accuracy

The recognition accuracy obtained by using the iris dataset is shown in Fig. 2. The horizontal axis is the number of incremental data and the vertical axis is recognition accuracy. In the figure, the lines labeled “Improved ISPCA” and “ISPCA” are ones obtained by the improved Incremental Simple-PCA which is proposed in this paper and the previous Incremental Simple-PCA, respectively.

As a result, it is understood that the recognition accuracy that used the improved Incremental Simple-PCA proposed in this paper is higher than of that the previous Incremental Simple-PCA. Moreover, it turns out that it is closer to the result obtained using the batch Simple-PCA. We can explain this phenomenon from the degree of approximation of each eigenvector obtained by the batch Simple-PCA and the incremental learning. The inner product result of each eigenvector obtained by the batch Simple-PCA and the improved Incremental simple-PCA is shown in Fig. 3(a). In addition, Fig. 3(b) shows the result of the batch Simple-PCA and the previous Incremental Simple-PCA. These were

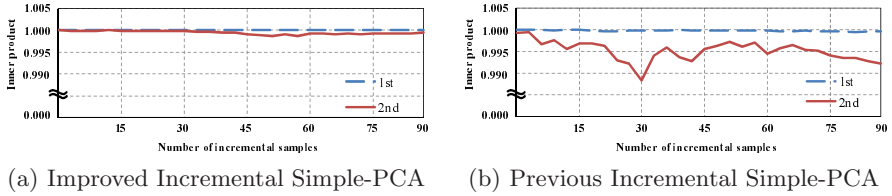


Fig. 3. The inner products between eigenvectors obtained by Batch SPCA and each incremental method

calculated by using the 1st and 2nd eigenvector every time the incremental sample is added. When the result of the inner product is the maximum value, that is 1, it means that each eigenvector found by the batch Simple-PCA and the Incremental algorithm is almost the same. In Fig. 3(a), both the inner product values of 1st and 2nd are always converged nearly to the maximum. On the other hand, in Fig. 3(b), it turns out that 2nd one deviates gradually as the incremental learning advances though the 1st eigenvector is almost approximated. Thus, the improved Incremental Simple-PCA can update eigenvector with high precision, and is more effective than the previous algorithm. Furthermore, the recognition accuracy when the Incremental PCA is used is low compared with Simple-PCA in Fig. 2. We think that the accuracy is varied by the difference of the property of the applied datasets. For the iris used in this experiment, it turns out that the Incremental Simple-PCA is more dominant than IPCA for the identification. Although the relative merits of these techniques are expected to vary in the viewpoint of the recognition accuracy by the problem to apply, it turns out that the Incremental Simple-PCA is more dominant for the identification problem with iris used in this experiment.

3.4 Computing Time

In this section, each technique is compared in relation to computational time. Fig. 4 shows the variation of the computation time that needed at the time of incremental learning. The vertical axis is computational time and the horizontal axis is the number of incremental data. As a result, the computing time is long in order of the improved Incremental Simple-PCA, the batch Simple-PCA, and the previous Incremental Simple-PCA. Though the previous Incremental Simple-PCA was the algorithm which updates the eigenvector by one-time calculation, the improved algorithm needs the number of times of update calculation which is the same as the number of times of calculation which were repeated by Simple-PCA in the initial learning. Therefore, computational time becomes longer by using the improved algorithm. On the other hand, it is observed that the computational time is very little when the Incremental PCA which needs matrix calculation is used. This is partly because the dimension of the applied dataset is very small. Actually, when higher dimensional data like facial images and the EMG signal, etc. is treated, it is shown that the Simple-PCA has been more high-speed [13].

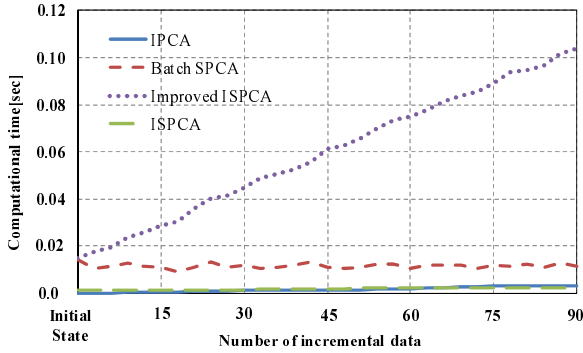


Fig. 4. Change in computing time

3.5 Memory Usage

Next, we consider the proposed algorithm from the standpoint of the space complexity. The proposal algorithm must hold $f_n^k (n = 1, \dots, L; k = 1, \dots, K_n)$ samples in order to update the eigenvectors. L and K_n are defined as the maximum number of eigenvector and the repeat count of the calculation required in order to obtain n -th eigenvector, respectively. The total number of repeated calculation K by Simple-PCA is represented $K = \sum_{n=1}^L K_n$. Thus, the storage region $K \cdot D$ is needed to execute the incremental learning by using improved Incremental Simple-PCA, where D is the number of dimensions of the samples. Hence, repeat count K that Simple-PCA needed in initial learning influences greatly the storage capacity of the proposed algorithm. In addition, when we define the number of input samples as N , the storage capacity which is required to execute the batch learning by using Simple-PCA is $N \cdot D$.

When the input samples N are few, which means $N < K$, $N \cdot D < K \cdot D$ is true. Therefore, the memory usage decreases by executing Simple-PCA to all the input samples as batch learning rather than performing incremental learning that used improved Incremental Simple-PCA. However, in the stage with many input samples ($N > K$), it can be said that the incremental learning with proposal algorithm is more effective than the batch learning in respect to memory usage. As stated above, the superiority and inferiority of both methods change according to the relationship between the number of input samples and the repetition count. Hence, it is necessary to judge which of the batch learning or the incremental learning should be used.

The total number of calculation repeated ($K = \sum_{n=1}^L K_n$) for obtaining each eigenvector using Simple-PCA is shown in Fig. 5. The vertical axis is the total repeat count of calculation K , and the horizontal axis is the number of input samples N . The dotted line in Fig. 5 represents the relationship when the number of input samples is equal to the total number of repeated calculation ($N = K$). The intersection of the dotted line and the solid line which shows repetition count is thought to become criterion for judgment of whether to use the incremental

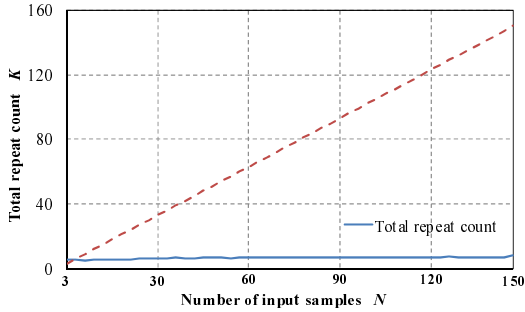


Fig. 5. Change in total frequency of repeat calculation

learning. In this figure, when the number of input samples is about 6, the two lines cross. In this case, when the number of input samples exceeds 6, it can be said that it is better to perform the incremental learning by using the proposal technique because the memory utilization is less. Thus, the incremental learning is effective unless the number of input samples is few. In addition, the incremental learning which used the proposal algorithm is indispensable when the batch samples cannot be obtained.

4 Conclusion

In this paper, the algorithm which made improvement to Incremental Simple-PCA which added the incremental learning function to Simple-PCA which is an approximation algorithm of the principal component analysis was proposed. Because the previous Incremental Simple-PCA has a problem that related to the convergency of the eigenvector, we tried the improvement of this point. As a result of the verification experiment, the effectiveness was able to be confirmed by viewpoints of the recognition accuracy and the convergency of the eigenvector. However, it was found that the computing time is much longer than Incremental PCA which needs matrix calculation and previous Incremental Simple-PCA. Moreover, the memory usage increases because it needs information more than the previous algorithm when the eigenvector is updated. Therefore, further improvement is needed. In addition, more detailed verification remains to be done by applying it to other various recognition problems, such as facial recognition.

References

1. Sakano, H., Yamada, K.: Horror Story: The Curse of Dimensionality. *Journal of Information Processing Society of Japan* 43(5), 562–567 (2002) (in Japanese)
2. Turk, M., Pentland, A.: Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 3(1), 71–86 (1991)
3. Nayar, S.K., Nene, S.A., Murase, H.: Subspace Methods for Robot Vision. *IEEE Trans. Robotics and Automation* 12(5), 750–758 (1996)

4. Murase, H., Nayar, S.K.: Visual Learning and Recognition of 3D Objects from Appearance. *International Journal of Computer Vision* 14, 5–24 (1995)
5. Hall, P.M., Marshall, D., Martin, R.R.: Incremental Eigenanalysis for Classification. In: *Proc. of the British Machine Vision Conference*, vol. 1, pp. 286–295 (1998)
6. Artac, M., Jogan, M., Leonardis, A.: Mobile robot localization using an incremental eigenspace model. In: *Proc. of IEEE International Conference on Robotics and Automation*, Washington, D.C., pp. 1025–1030 (2002)
7. Freitas, R., Santos-Victor, J., Sarcinelli-Filho, M., Bastos-Filho, T.: Performance Evaluation of Incremental Eigenspace Models for Mobile Robot Localization. In: *Proceedings of the IEEE 11th International Conference on Advanced Robotics (ICAR 2003)*, Coimbra, Portugal, pp. 417–422 (2003)
8. Artac, M., Jogan, M., Leonardis, A.: Incremental PCA for On-line Visual Learning and Recognition. In: *Proceedings of the 16th International Conference on Pattern Recognition (ICPR)*, Quebec City, Canada, pp. 781–784 (2002)
9. Partridge, M., Calvo, R.: Fast dimensionality reduction and simple PCA. In: *IDA*, vol. 2, pp. 292–298 (1997)
10. Kuroiwa, S., Tsuge, S., Tani, H., Tai, X.-Y., Shishibori, M., Kita, K.: Dimensionality reduction of vector space model based on Simple PCA. In: *Proc. Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies (KES)*, Osaka, vol. 2, pp. 362–366 (2001)
11. Nakano, M., Yasukata, F., Fukumi, M.: Recognition of Smiling Faces Using Neural Networks and SPCA. *International Journal of Computational Intelligence and Applications* 4(2), 153–164 (2004)
12. Takimoto, H., Mitsukura, Y., Fukumi, M., Akamatsu, N.: A Feature Extraction Method for Personal Identification System by Using Real-Coded Genetic Algorithm. In: *Proc. of 7th SCI 2003*, Orlando, USA, vol. 4, pp. 66–77 (2003)
13. Oyama, T., Karungaru, S.G., Tsuge, S., Mitsukura, Y., Fukumi, M.: Fast Incremental Algorithm of Simple Principal Component Analysis. *IEEJ Trans. on Electronics, Information and Systems* 129(1), 112–117 (2009)
14. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2007), <http://www.ics.uci.edu/~mllearn/MLRepository.html>

A Meta-learning Method Based on Temporal Difference Error

Kunikazu Kobayashi, Hiroyuki Mizoue, Takashi Kuremoto, and Masanao Obayashi

Yamaguchi University, Tokiwadai 2-16-1, Ube, Yamaguchi 755-8611, Japan
{koba, wu, m.obayas}@yamaguchi-u.ac.jp
<http://www.nn.csse.yamaguchi-u.ac.jp/k/>

Abstract. In general, meta-parameters in a reinforcement learning system, such as a learning rate and a discount rate, are empirically determined and fixed during learning. When an external environment is therefore changed, the system cannot adapt itself to the variation. Meanwhile, it is suggested that the biological brain might conduct reinforcement learning and adapt itself to the external environment by controlling neuromodulators corresponding to the meta-parameters. In the present paper, based on the above suggestion, a method to adjust meta-parameters using a temporal difference (TD) error is proposed. Through various computer simulations using a maze search problem and an inverted pendulum control problem, it is verified that the proposed method could appropriately adjust meta-parameters according to the variation of the external environment.

Keywords: reinforcement learning, meta-parameter, meta-learning, TD-error, maze search problem, inverted pendulum control problem.

1 Introduction

Reinforcement learning is a famous model of animal learning [1]. Schultz et al. found that dopamine neurons in the basal ganglia have the formal characteristics of the teaching signal known as the temporal difference (TD) error through an animal experiment [2].

In this context, Doya proposed the hypotheses between meta-parameters in reinforcement learning and neuromodulators in the basal ganglia based on the review of experimental data and theoretical models. That is, he presented that dopamine signals the error in reward prediction, serotonin controls the time scale of reward prediction, noradrenaline controls the randomness in action selection, and acetylcholine controls the speed of memory update [3]. Successful reinforcement learning highly depends on the careful setting of meta-parameters in reinforcement learning. Schweighofer et al. proposed a meta-learning method based on rewards, which not only finds appropriate meta-parameters but also controls the time course of these meta-parameters in an adaptive manner [4]. However, their method has some parameters to be pre-determined.

In the present paper, we propose a meta-learning method based on a TD-error. The proposed method has only one parameter to be pre-determined and is easy to apply to reinforcement learning. Through various computer simulations using a maze search problem and an inverted pendulum control problem, it is verified that the proposed

method allows meta-parameters to be appropriately adjusted according to the variation of the external environment.

2 Reinforcement Learning

Reinforcement learning is a method that agents acquire the optimum behavior with a repeating process of exploration and exploitation by being given rewards in an environment as a compensation for its behavior [1]. In this section, we explain two kinds of temporal difference (TD) learning, i.e. a Q-learning method [5] and an actor-critic method [1] and also describe a policy in reinforcement learning.

2.1 Q-Learning Method

The Q-learning method guarantees that every state converges to the optimal solution by appropriately adjusting a learning rate in an MDP environment [5]. The state-action value function $Q(s(t), a(t))$ for a state $s(t)$ at time t and an action $a(t)$ at time t is updated so as to take the optimal action by exploring it in a learning space and defined as follows:

$$Q(s(t), a(t)) \leftarrow Q(s(t), a(t)) + \alpha \delta(t), \quad (1)$$

$$\delta(t) = r(t) + \gamma \max_{b \in A} Q(s(t+1), b) - Q(s(t), a(t)), \quad (2)$$

where $\delta(t)$ and $r(t)$ denote a TD-error at time t and a reward at time t , respectively, meta-parameters α and γ refer to a learning rate and a discount rate, respectively, and A is a set of actions to be taken.

2.2 Actor-Critic Method

The actor-critic method has a separate memory structure to explicitly represent the policy independent of the value function [1]. The policy structure is known as the actor, because it is used to select actions, and the estimated value function is known as the critic, because it criticizes the actions made by the actor. The values in actor-critic method are updated as follows:

$$V(s(t)) \leftarrow V(s(t)) + \alpha \delta(t), \quad (3)$$

$$Q(s(t), a(t)) \leftarrow Q(s(t), a(t)) + \alpha \delta(t), \quad (4)$$

$$\delta(t) = r(t) + \gamma V(s(t+1)) - V(s(t)), \quad (5)$$

where $V(s(t))$ is a value function for a state $s(t)$ at time t .

2.3 Policy

The policy is a mapping from the states in an external environment to the actions to take in those states. Throughout the present paper, we suppose that an action is selected by the Boltzmann distribution. That is, the policy $\pi(s(t), a(t))$ is defined as follows [1]:

$$\pi(s(t), a(t)) = \frac{\exp(Q(s(t), a(t))/T)}{\sum_{b \in A} \exp(Q(s(t), b)/T)}, \quad (6)$$

where T refers to a temperature parameter. The policy realizes a random selection if $T \rightarrow \infty$ and is a greedy selection if $T \rightarrow 0$.

3 Meta-learning

Generally speaking, it is crucial that all the meta-parameters such as a learning rate and a discount rate are carefully tuned to elicit good performance in advance. It therefore is beneficial that the meta-parameters could be changed according to the situation. In this section, we describe the conventional meta-learning method based on rewards (Section 3.1) and propose a new meta-learning method based on the TD-error (Section 3.2).

3.1 Meta-learning Based on Reward

Schweighofer et al. proposed the meta-learning method based on mid-term and long-term rewards [4]. In their method, the mid-term reward $r_{MT}(t)$ at time t and the long-term reward $r_{LT}(t)$ at time t are defined as follows:

$$r_{MT}(t) = \left(1 - \frac{1}{\tau_{MT}}\right) r_{MT}(t-1) + r(t), \quad (7)$$

$$r_{LT}(t) = \left(1 - \frac{1}{\tau_{LT}}\right) r_{LT}(t-1) + r_{MT}(t), \quad (8)$$

where $r(t)$ refers to an instant reward at time t , τ_{MT} and τ_{LT} denote the time constants for $r_{MT}(t)$ and $r_{LT}(t)$, respectively. In the present paper, we assume that $\tau_{MT} < \tau_{LT}$ and $r_{MT}(0) = r_{LT}(0) = 0$. If an agent tends to take the desired actions than before, then the mid-term reward has a larger value than the long-term one. If not so, the mid-term reward has a smaller value than the long-term one. The conventional method updates meta-parameters such as a discount rate, a learning rate, and a temperature parameter using this characteristic.

A discount rate is defined as a function of time t .

$$\gamma(t) = 1 - e^{-\epsilon(t)}, \quad (9)$$

where $\epsilon(t)$ refers to a variable represented by the rewards and an exploration noise $\sigma(t)$, and is defined as follows:

$$\epsilon(t) = \epsilon'(t) + \sigma(t). \quad (10)$$

In [10], $\epsilon'(t)$ is a variable depending on the rewards and updated by the following equation ($\epsilon'(0) = 0$).

$$\epsilon'(t) = \epsilon'(t-1) + \mu \{r_{MT}(t) - r_{LT}(t)\} \sigma(t), \quad (11)$$

where μ represents an updating rate for $\epsilon'(t)$.

Although the other meta-parameters, i.e. a learning rate $\alpha(t)$ and a temperature parameter $T(t)$, are also updated like the above, the literature [4] does not present their

updating rules at all. In the present paper, we therefore propose the updating rules for $\alpha(t)$ and $T(t)$. At first, we propose that $\alpha(t)$ and $T(t)$ are defined as follows:

$$\alpha(t) = e^{-\epsilon(t)} \tag{12}$$

$$T(t) = \frac{1}{e^{\epsilon(t)} - 1} \tag{13}$$

From (10), the value of $\epsilon(t)$ increases if $r_{MT}(t)$ is larger than $r_{LT}(t)$, decreases if $r_{MT}(t) < r_{LT}(t)$, and has no change if $r_{MT}(t) \approx r_{LT}(t)$. The $\alpha(t)$ and $T(t)$ therefore increases if $r_{MT}(t) > r_{LT}(t)$ because of the growth of $\epsilon(t)$ and decreases if not so. This corresponds that $\alpha(t)$ and $T(t)$ should take large values if learning is required, e.g. in the beginning of learning or when the external environment has changed, and they should take small values if not so.

In the present paper, it is assumed that the conventional method includes the two proposed equations (12) and (13) besides the conventional equation (9) in the literature [4]. Note that the conventional method does not completely correspond with Schweighofer et al.’s method.

3.2 Meta-learning Based on TD-Error

The optimal values of meta-parameters in reinforcement learning might change according to the progress of learning. We therefore focus on the TD-error which could be changed by the progress of learning and propose a meta-learning method based on the TD-error. We define a variable $\delta'(t)$ which depends on the absolute value of the TD-error. Then, meta-parameters are updated based on it. The $\delta'(t)$ is defined as follows ($\delta'(0) = 0$):

$$\delta'(t) = \left(1 - \frac{1}{\tau}\right) \delta'(t-1) + \frac{1}{\tau} |\delta(t)|, \tag{14}$$

where $\delta(t)$ refers to a TD-error at time t and τ is a time constant.

A learning rate $\alpha(t)$ and a temperature parameter $T(t)$ are expected to be a large value for exploration in the beginning of learning. On the other hand, their values are desired to be small for exploitation at the matured stage of learning. In addition, if relearning is required according to an environmental change, a discount rate $\gamma(t)$ are expected to be a small value. On the other hand, the value of $\delta'(t)$ becomes large if relearning is required because of the environmental change and converges to 0 at the matured stage. The meta-parameters $\alpha(t)$, $\gamma(t)$, and $T(t)$ at time t are therefore defined based on $\delta'(t)$ as follows:

$$\alpha(t) = \frac{2}{1 + e^{-\delta'(t)}} - 1, \tag{15}$$

$$\gamma(t) = \frac{2}{1 + e^{\delta'(t)}}, \tag{16}$$

$$T(t) = e^{\delta'(t)} - 1. \tag{17}$$

Based on the above equations, $\alpha(t)$ and $T(t)$ becomes small according to the decrease of $\delta'(t)$ and becomes large according to the increase of $\delta'(t)$. This allows that meta-parameters are appropriately updated in accordance with the change of $\delta'(t)$.

4 Computer Simulation

The performance of the proposed method is verified through computer simulation. In the simulation, we prepare two reinforcement learning tasks, i.e. a maze search problem and an inverted pendulum control problem. Accordingly, we apply the proposed method to the Q-learning method (Section 2.1) and the actor-critic method (Section 2.2). The proposed method applied to the Q-learning system is compared with the conventional method (Section 3.1) and the Q-learning without meta-learning (Section 2.1). Then, our method applied to the actor-critic system is compared with the conventional method and the actor-critic method without meta-learning (Section 2.2).

4.1 Maze Search Problem

To evaluate the performance for the discrete task, we use the maze search problem. In this simulation, we apply the proposed method to the Q-learning system.

Simulation Setting. Figure 1 shows a maze used in the simulation. In this figure, the black and gray squares correspond to walls and the white squares correspond to paths. The structure of the maze in Fig. 1(a) will change to that in Fig. 1(b) at 301 episodes. Namely, one gray square and two white squares are changed to a white one and two gray ones, respectively. The shortest path for both mazes is 14 steps. An agent is able to perceive only the adjacent eight squares. In this task, the Markov property is guaranteed because there is no aliasing problem. But, since the structure of the maze is changed, meta-parameters should be adjusted again. In this simulation, one episode is assumed that an agent starts from a start point and arrives at a goal point. The failure of maze search is defined when an agent cannot arrive at a goal point within 10,000 steps.

The parameters to be pre-determined are as follows: In the conventional method, time constant in (7) and (8): $\tau_{MT} = \tau_{LT} = 300$, updating rate in (11): $\mu = 0.1$. In the proposed method, time constant in (14): $\tau = 300$. The exploration noise $\sigma(t)$ in (10) is set as the Gaussian distribution with mean 0 and variance 1. In the standard method with fixed meta-parameters, we set as $\alpha = 0.2$, $\gamma = 0.95$, and $T = 0.3$.

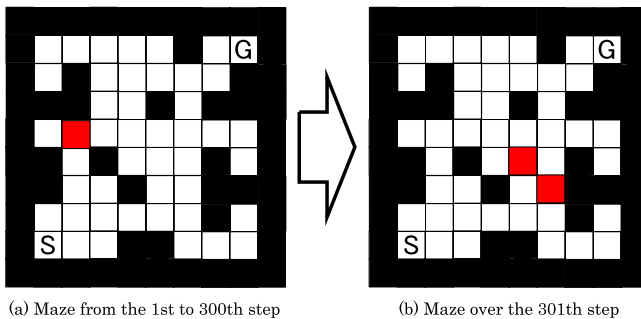
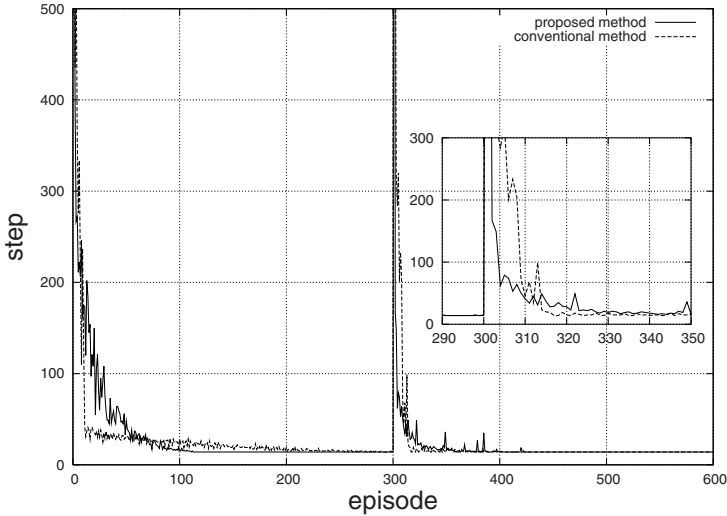


Fig. 1. Dynamical structure change in the maze search problem

Table 1. The total number of steps in the maze search problem

Method	From 1 to 300 episodes	From 1 to 600 episodes
Proposed method	9,855	17,622
Conventional method	9,922	23,226
Q-learning	10,664	—

**Fig. 2.** Development of the number of steps for the conventional and proposed methods in the maze search problem

Simulation Result. Table 1 shows the total number of steps and Fig 2 illustrates the development of the number of steps. From these results, the proposed method could adjust to the structure change of the maze. That is, although the number of steps increases significantly when the structure is changed at 301 steps, both the proposed and conventional methods can find a new shortest path but the Q-learning method without meta-learning cannot find it. Furthermore, as seen from Table 1, the proposed method takes smaller steps than the conventional method.

4.2 Inverted Pendulum Control Problem

To evaluate the performance for the continuous task, we use the inverted pendulum control problem. In this simulation, we apply the proposed method to the actor-critic system.

Simulation Setting. Let θ , $\dot{\theta}$ ($= d\theta/dt$), and τ_c be angle, angular velocity, and torque, respectively. The dynamics of the inverted pendulum is represented by

$$ml^2\ddot{\theta} = -mgl \sin(\theta) - \mu\dot{\theta} + \tau_c, \quad (18)$$

where m denotes the mass of the pendulum, l is the length of the pendulum, g is the acceleration of gravity, μ is the friction coefficient of axis. In the simulation, we set $m = 0.5[kg]$, $l = 0.5[m]$, $g = 9.8[m/s^2]$, and $\mu = 0.1$. Then, reward $r(t)$ is defined as follows:

$$r(t) = \cos(\theta) - 0.005 \tau_c^2(t). \quad (19)$$

At the initial state, we set $\theta(0) = 0$ and $\dot{\theta}(t) = 0$. We can only observe θ and set the torque to the pendulum as -1 or $+1$. The success of control is assumed that the pendulum can be controlled within $\pm 5[deg]$ for $15[s]$. Then, the trial that the angle of the pendulum exceeds $\pm 45[deg]$ is assumed to be failure and go to the next trial. A time step is set as $\Delta t = 0.01[s]$. The success rate of controlling the pendulum is calculated as the average of 100 trials after 1,000 training episodes.

The parameters are set as $\tau_{MT} = \tau_{LT} = 300$ and $\mu = 0.1$ in the conventional method, and $\tau = 300$ in the proposed method. The exploration noise is set as the Gaussian distribution with mean 0 and variance 1. In the standard method with fixed meta-parameters, we set as $\alpha = 0.2$, $\gamma = 0.95$, and $T = 0.35$.

Simulation Result. As seen in Table 2, the success rate is 87.9% in the proposed method, 85.2% in the conventional method, and 76.8% in Q-learning method. As a result, it is shown that the proposed method could improve the performance for controlling the pendulum. Figure 3 shows the temporal development of the accumulated

Table 2. The success rate of controlling an inverted pendulum

Method	Success rate (%)
Proposed method	87.9
Conventional method	85.2
Actor-critic method	76.8

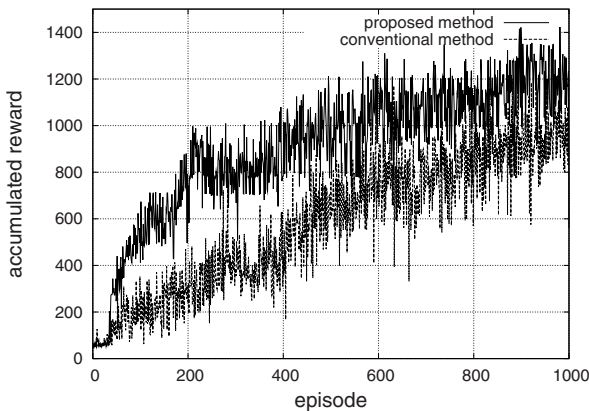


Fig. 3. Development of the accumulated reward for the proposed and conventional methods in the inverted pendulum control problem

reward. From this figure, it is clear that the accumulated reward in the proposed method is much larger than that in the conventional method. In addition, the number of parameters to be pre-determined is only one, i.e. τ in (14) in the proposed method. On the other hand, there are three such parameters, i.e. τ_{MT} in (7), τ_{LT} in (8), and μ in (11) in the conventional method. It therefore is much easier for the proposed method to apply the reinforcement learning system than the conventional method.

5 Summary

The present paper have proposed the meta-learning method based on the TD-error. Through various computer simulations, we investigated the performance of the proposed method using the discrete and continuous tasks. As a result, it is shown that the proposed method applied to the Q-learning system could improve the learning performance for the discrete task, the maze search problem because it allows meta-parameters to adjust according to the variation of the external environment. In addition, it is clarified that the proposed method applied to the actor-critic system could improve the control performance for the continuous task, the inverted pendulum control problem. Furthermore, it is shown that the proposed method could easily apply to reinforcement learning compared with the conventional method, the standard Q-learning and actor-critic methods because the proposed method can reduce the number of parameters to be pre-determined. In future work, the proposed method under a noisy environment needs to be evaluated.

Acknowledgments. This work was partly supported by three Grant-in-Aid projects for Scientific Research (No.20500207 and 20500277) from MEXT, Japan.

References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
2. Schultz, W., Dayan, P., Montague, P.R.: A Neural Substrate of Prediction and Reward. *Science* 275, 1593–1599 (1997)
3. Doya, K.: Metalearning and Neuromodulation. *Neural Networks* 15, 495–506 (2002)
4. Schweighofer, N., Doya, K.: Meta-learning in Reinforcement Learning. *Neural Networks* 16(1), 5–9 (2003)
5. Watkins, C.J.C.H., Dayan, P.: Q-learning. *Machine Learning* 8(3-4), 279–292 (1992)
6. Ishii, S., Yoshida, W., Yoshimoto, J.: Control of Exploitation-Exploration Meta-parameter in Reinforcement Learning. *Neural Networks* 15(4-6), 665–687 (2002)

Local Learning Rules for Nonnegative Tucker Decomposition

Anh Huy Phan and Andrzej Cichocki*

Lab for Advanced Brain Signal Processing
Brain Science Institute - RIKEN
Wako-shi, Saitama 351-0198, Japan
{phan,cia}@brain.riken.jp

Abstract. Analysis of data with high dimensionality in modern applications, such as spectral analysis, neuroscience, chemometrics naturally requires tensorial approaches different from standard matrix factorizations (PCA, ICA, NMF). The Tucker decomposition and its constrained versions with sparsity and/or nonnegativity constraints allow for the extraction of different numbers of hidden factors in each of the modes, and permits interactions within each modality having many potential applications in computational neuroscience, text mining, and data analysis. In this paper, we propose a new algorithm for Nonnegative Tucker Decomposition (NTD) based on a constrained minimization of a set of local cost functions which is suitable for large scale problems. Extensive experiments confirm the validity and high performance of the developed algorithms in comparison with other well-known algorithms.

1 Introduction

In many applications such as those in neuroscience studies, the data structures often contain high-order ways (modes) including trials, task conditions, subjects, together with the intrinsic dimensions of space, time, and frequency. Analysis on separate matrices or slices extracted from a data block often faces the risk of losing the covariance information among subjects. To discover hidden components within the data, the analysis tools should reflect the multi-dimensional structure of the data [1].

Tucker decomposition is a suitable method to extract factors having interactive relations within each modality, described as a “decomposition of a given N -th order tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ into an unknown core tensor $\underline{\mathbf{G}} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ (typically $J_n \ll I_n$) multiplied by a set of N unknown component matrices, $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \mathbf{a}_2^{(n)}, \dots, \mathbf{a}_{J_n}^{(n)}] \in \mathbb{R}^{I_n \times J_n}$ ($n = 1, 2, \dots, N$), representing common factors or loadings” [2][3].

$$\underline{\mathbf{Y}} = \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \dots \sum_{j_N=1}^{J_N} g_{j_1 j_2 \dots j_N} \mathbf{a}_{j_1}^{(1)} \circ \mathbf{a}_{j_2}^{(2)} \circ \dots \circ \mathbf{a}_{j_N}^{(N)} + \underline{\mathbf{E}}$$

* Also from Dept. EE Warsaw University of Technology and Systems Research Institute, Polish Academy of Science, Poland.

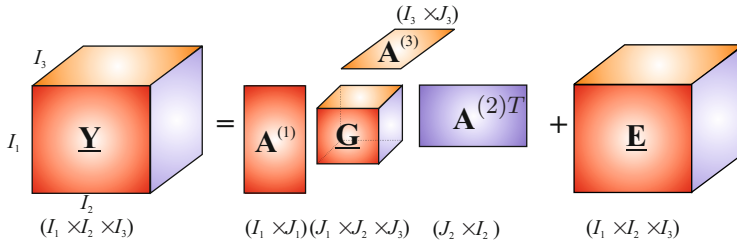


Fig. 1. Illustration for a third-order Tucker decomposition; the objective here is to find optimal component matrices $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times J_n}$ and a core tensor $\mathbf{G} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$

$$\begin{aligned}
 &= \underline{\mathbf{G}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} + \underline{\mathbf{E}} \\
 &= \underline{\mathbf{G}} \times \{\mathbf{A}\} + \underline{\mathbf{E}} = \widehat{\mathbf{Y}} + \underline{\mathbf{E}},
 \end{aligned} \tag{1}$$

where $\widehat{\mathbf{Y}}$ is an approximation of \mathbf{Y} , and $\underline{\mathbf{E}}$ denotes the estimation error (see Fig. 1 as an example of a 3-way Tucker decomposition). Sparsity and nonnegativity constraints often imposed on hidden factors and core tensor due to meaningful representation leads to the NTD model with many potential applications in neuroscience, bioinformatics, chemometrics ect [14].

Almost all the existing algorithms for Tucker decompositions [3, 4, 5] require processing based on full tensor during the estimation. The real-world data often contain millions of elements. Full data processing, especially inverse of huge matrices, are therefore impractical. To this end, we formulate local learning rules which sequentially estimate components in each factors. The proposed algorithm called the Hierarchical Alternative Least Square (HALS) algorithm has the successful original forms for NMF and PARAFAC models [16].

Extensive experiments confirm the validity and high performance of the developed algorithms on the applications of noisy data reconstruction, classification of EEG data. The performance of new algorithm was compared to well-known existing algorithms (HOOI [3], HONMF [4]).

2 Local Tucker Decomposition

Most algorithms for the NTD model are based on ALS minimization of the squared Euclidean distance [3, 7] used as a global cost function subject to non-negativity constraints, that is

$$D_F(\mathbf{Y} \parallel \underline{\mathbf{G}} \times \{\mathbf{A}\}) = \frac{1}{2} \left\| \mathbf{Y} - \widehat{\mathbf{Y}} \right\|_F^2. \tag{2}$$

Based upon some adjustments on this cost function, we establish local learning rules for components and core tensor.

¹ For convenience, tensor notations used in this paper are adopted from [3].

2.1 Learning Rule for Factors $\mathbf{A}^{(n)}$

We define the residual tensor $\underline{\mathbf{Y}}^{(j_n)}$

$$\begin{aligned} \underline{\mathbf{Y}}^{(j_n)} &= \underline{\mathbf{Y}} - \sum_{r_1=1}^{J_1} \dots \sum_{r_n \neq j_n} \dots \sum_{r_N=1}^{J_N} g_{r_1 \dots r_n \dots r_N} \mathbf{a}_{j_1}^{(1)} \circ \dots \circ \mathbf{a}_{r_n}^{(n)} \circ \dots \circ \mathbf{a}_{r_N}^{(N)} \\ &= \underline{\mathbf{Y}} - \widehat{\underline{\mathbf{Y}}} + \underline{\mathbf{G}}_{r_n=j_n} \times_{-n} \{\mathbf{A}\} \times_n \mathbf{a}_{j_n}^{(n)}, \quad (j_n = 1, 2, \dots, J_N), \end{aligned} \quad (3)$$

where $\underline{\mathbf{G}}_{r_n=j_n} \in \mathbb{R}^{J_1 \times \dots \times J_{n-1} \times 1 \times J_{n+1} \times \dots \times J_N}$ is a subtensor of the tensor $\underline{\mathbf{G}}$ obtained by fixing the n -th index to some value j_n . For example, for a three-way core tensor $\underline{\mathbf{G}} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$, $\underline{\mathbf{G}}_{r_1=1}$ is the first horizontal slice, but of size $1 \times J_2 \times J_3$. The mode- n matricized version of tensor $\underline{\mathbf{G}}_{r_n=j_n}$ is exactly the j_n -th row of the mode- n matricized version of tensor $\underline{\mathbf{G}}$, i.e., $[\underline{\mathbf{G}}_{r_n=j_n}]_{(n)} = [\mathbf{G}^{(n)}]_{j_n}$.

To estimate the component $\mathbf{a}_{j_n}^{(n)}$, we assume that all the other components in all factors and the core tensor are fixed. Instead of minimizing (2), we use a more sophisticated approach by minimizing a set of local cost functions given by:

$$\begin{aligned} D_F^{(j_n)} &= \frac{1}{2} \left\| \underline{\mathbf{Y}}^{(j_n)} - \sum_{\substack{r_1, \dots, r_{n-1}, \\ r_{n+1}, \dots, r_N}} g_{r_1 \dots r_{n-1} j_n r_{n+1} \dots r_N} \right. \\ &\quad \left. \mathbf{a}_{r_1}^{(1)} \circ \dots \circ \mathbf{a}_{r_{n-1}}^{(n-1)} \circ \mathbf{a}_{j_n}^{(n)} \circ \mathbf{a}_{r_{n+1}}^{(n+1)} \circ \dots \circ \mathbf{a}_{r_N}^{(N)} \right\|_F^2 \\ &= \frac{1}{2} \left\| \underline{\mathbf{Y}}^{(j_n)} - \underline{\mathbf{G}}_{r_n=j_n} \times_{-n} \{\mathbf{A}\} \times_n \mathbf{a}_{j_n}^{(n)} \right\|_F^2 \\ &= \frac{1}{2} \left\| \mathbf{Y}^{(j_n)}_{(n)} - \mathbf{a}_{j_n}^{(n)} [\underline{\mathbf{G}}_{r_n=j_n}]_{(n)} \mathbf{A}^{\otimes -n T} \right\|_F^2, \end{aligned} \quad (4)$$

We first calculate the gradient of (4) with respect to element $\mathbf{a}_{j_n}^{(n)}$

$$\frac{\partial D_F^{(j_n)}}{\partial \mathbf{a}_{j_n}^{(n)}} = - \left(\mathbf{Y}^{(j_n)}_{(n)} - \mathbf{a}_{j_n}^{(n)} [\underline{\mathbf{G}}_{(n)}]_{j_n} \mathbf{A}^{\otimes -n T} \right) \mathbf{A}^{\otimes -n} [\underline{\mathbf{G}}_{(n)}]_{j_n}^T \quad (5)$$

and set it to zero to obtain a fixed point learning rule for $\mathbf{a}_{j_n}^{(n)}$ given by

$$\mathbf{a}_{j_n}^{(n)} \leftarrow \mathbf{Y}^{(j_n)}_{(n)} \mathbf{A}^{\otimes -n} [\underline{\mathbf{G}}_{(n)}]_{j_n}^T / \left([\underline{\mathbf{G}}_{(n)}]_{j_n} \mathbf{A}^{\otimes -n T} \mathbf{A}^{\otimes -n} [\underline{\mathbf{G}}_{(n)}]_{j_n}^T \right), \quad (6)$$

for $n = 1, 2, \dots, N$ and $j_n = 1, 2, \dots, J_N$. In the next step we shall further optimize the derived learning rules.

2.2 Update Rules for the Core Tensor

Elements of the core tensor will be sequentially updated under the assumption that all components are fixed. The cost function (4) is adjusted as follows

$$D_F^{(j_n)} = \frac{1}{2} \left\| \underline{\mathbf{Y}}^{(j_n)} - \sum_{\substack{(r_1, \dots, r_{n-1}, r_{n+1}, \dots, r_N) \neq \\ (j_1, \dots, j_{n-1}, j_{n+1}, \dots, j_N)}} (g_{r_1 \dots r_{n-1} j_n r_{n+1} \dots r_N} \mathbf{a}_{r_1}^{(1)} \circ \dots \circ \mathbf{a}_{r_{n-1}}^{(n-1)} \circ \dots \circ \mathbf{a}_{r_{n+1}}^{(n+1)} \circ \dots \circ \mathbf{a}_{r_N}^{(N)}) \right\|_F^2$$

$$\begin{aligned}
 & \mathbf{a}_{j_n}^{(n)} \circ \mathbf{a}_{r_{n+1}}^{(n+1)} \circ \dots \circ \mathbf{a}_{r_N}^{(N)} - g_{j_1 \dots j_n \dots j_N} \mathbf{a}_{j_1}^{(1)} \circ \dots \circ \mathbf{a}_{j_n}^{(n)} \circ \dots \circ \mathbf{a}_{j_N}^{(N)} \Big\|_F^2 \\
 = & \frac{1}{2} \Big\| \underline{\mathbf{Y}} - \sum_{\substack{(r_1, \dots, r_n, \dots, r_N) \neq \\ (j_1, \dots, j_n, \dots, j_N)}} g_{r_1 \dots r_n \dots r_N} \mathbf{a}_{r_1}^{(1)} \circ \dots \circ \mathbf{a}_{r_n}^{(n)} \circ \dots \circ \mathbf{a}_{r_N}^{(N)} \\
 & - g_{j_1 \dots j_n \dots j_N} \mathbf{a}_{j_1}^{(1)} \circ \dots \circ \mathbf{a}_{j_n}^{(n)} \circ \dots \circ \mathbf{a}_{j_N}^{(N)} \Big\|_F^2 \\
 = & \frac{1}{2} \Big\| \underline{\mathbf{Y}}^{(\bar{j})} - g_{\bar{j}} \mathbf{a}_{j_1}^{(1)} \circ \dots \circ \mathbf{a}_{j_N}^{(N)} \Big\|_F^2, \tag{7}
 \end{aligned}$$

where the tensor $\underline{\mathbf{Y}}^{(\bar{j})}$, $\bar{j} = [j_1, j_2, \dots, j_N]$ is defined as

$$\begin{aligned}
 \underline{\mathbf{Y}}^{(\bar{j})} &= \underline{\mathbf{Y}} - \sum_{r_1 \neq j_1} \dots \sum_{r_N \neq j_N} g_{\bar{j}} \mathbf{a}_{r_1}^{(1)} \circ \dots \circ \mathbf{a}_{r_N}^{(N)} \\
 &= \underline{\mathbf{Y}} - \widehat{\underline{\mathbf{Y}}} + g_{\bar{j}} \mathbf{a}_{j_1}^{(1)} \circ \dots \circ \mathbf{a}_{j_N}^{(N)} = \underline{\mathbf{E}} + g_{\bar{j}} \mathbf{a}_{j_1}^{(1)} \circ \dots \circ \mathbf{a}_{j_N}^{(N)}. \tag{8}
 \end{aligned}$$

To estimate an entry $g_{\bar{j}}$, we consider the vectorized version of the cost function (7), and for convenience, we change the index of the cost function

$$D_F^{(\bar{j})} = \frac{1}{2} \Big\| \text{vec}(\mathbf{Y}_{(1)}^{(\bar{j})}) - \left(\mathbf{a}_{j_N}^{(N)} \otimes \dots \otimes \mathbf{a}_{j_1}^{(1)} \right) g_{\bar{j}} \Big\|_F^2. \tag{9}$$

The gradient of (9) with respect to element $g_{\bar{j}}$ is given by

$$\frac{\partial D_F^{(\bar{j})}}{\partial g_{\bar{j}}} = - \left(\mathbf{a}_{j_N}^{(N)} \otimes \dots \otimes \mathbf{a}_{j_1}^{(1)} \right)^T \left(\text{vec}(\mathbf{Y}_{(1)}^{(\bar{j})}) - \left(\mathbf{a}_{j_N}^{(N)} \otimes \dots \otimes \mathbf{a}_{j_1}^{(1)} \right) g_{\bar{j}} \right) \tag{10}$$

and set to zero, to yield learning rule for entries of the core tensor $\underline{\mathbf{G}}$

$$g_{\bar{j}} \leftarrow \left[\frac{\left(\mathbf{a}_{j_N}^{(N)} \otimes \dots \otimes \mathbf{a}_{j_1}^{(1)} \right)^T \text{vec}(\mathbf{Y}_{(1)}^{(\bar{j})})}{\left(\mathbf{a}_{j_N}^{(N)} \otimes \dots \otimes \mathbf{a}_{j_1}^{(1)} \right)^T \left(\mathbf{a}_{j_N}^{(N)} \otimes \dots \otimes \mathbf{a}_{j_1}^{(1)} \right)} \right]_+. \tag{11}$$

This update rule can be simplified by taking into account that the Kronecker product of the two unit-length vectors \mathbf{a} and \mathbf{b} , i.e., $\mathbf{c} = \mathbf{a} \otimes \mathbf{b}$, is also a unit-length vector, that is

$$\|\mathbf{c}\|_2^2 = \mathbf{c}^T \mathbf{c} = (\mathbf{a} \otimes \mathbf{b})^T (\mathbf{a} \otimes \mathbf{b}) = (\mathbf{a}^T \mathbf{a}) \otimes (\mathbf{b}^T \mathbf{b}) = 1 \otimes 1 = 1. \tag{12}$$

Hence, if all the components $\mathbf{a}_{j_n}^{(n)}$ are normalized to ℓ_2 unit length vectors, and by replacing $\mathbf{Y}_{(1)}^{(\bar{j})}$ in (11) by (8), we have

$$\begin{aligned}
 g_{\bar{j}} &\leftarrow \left(\mathbf{a}_{j_N}^{(N)} \otimes \dots \otimes \mathbf{a}_{j_1}^{(1)} \right)^T \text{vec} \left(\mathbf{E}_{(1)} + \left(\mathbf{a}_{j_N}^{(N)} \otimes \dots \otimes \mathbf{a}_{j_1}^{(1)} \right) g_{\bar{j}} \right) \\
 &= g_{\bar{j}} + \text{vec} \left(\mathbf{a}_{j_1}^{(1)T} \mathbf{E}_{(1)} \left(\mathbf{a}_{j_N}^{(N)T} \otimes \dots \otimes \mathbf{a}_{j_2}^{(2)T} \right)^T \right) \\
 &= g_{\bar{j}} + \underline{\mathbf{E}} \bar{\times}_1 \mathbf{a}_{j_1}^{(1)} \bar{\times}_2 \mathbf{a}_{j_2}^{(2)} \dots \bar{\times}_N \mathbf{a}_{j_N}^{(N)}. \tag{13}
 \end{aligned}$$

Moreover, the denominator of the learning rule (6) can also be neglected as it is a scale factor equal to one.

Finally, the learning rules for factors $\mathbf{A}^{(n)}$ ($n = 1, 2, \dots, N$) and core tensor $\underline{\mathbf{G}}$ can be summarized as follows (referred here to as the ℓ_2 HALS-NTD algorithm)

$$\mathbf{a}_{j_n}^{(n)} \leftarrow \left[\mathbf{Y}_{(n)}^{(j_n)} \left[(\underline{\mathbf{G}} \times_{-n} \{\mathbf{A}\})_{j_n} \right]^T \right]_+, \quad \mathbf{a}_{j_n}^{(n)} \leftarrow \mathbf{a}_{j_n}^{(n)} / \left\| \mathbf{a}_{j_n}^{(n)} \right\|_2, \quad (14)$$

$$g_{\bar{j}} \leftarrow \left[g_{\bar{j}} + \underline{\mathbf{E}} \bar{\times}_1 \mathbf{a}_{j_1}^{(1)} \bar{\times}_2 \mathbf{a}_{j_2}^{(2)} \cdots \bar{\times}_N \mathbf{a}_{j_N}^{(N)} \right]_+, \quad (j_n = 1, 2, \dots, J_N), \quad (15)$$

$$\underline{\mathbf{Y}}^{(j_n)} = \underline{\mathbf{E}} + g_{\bar{j}} \mathbf{a}_{j_1}^{(1)} \circ \cdots \circ \mathbf{a}_{j_N}^{(N)}. \quad (16)$$

The error tensor and the residue tensor don't need to be explicitly compute as in (1), and (3). They can be updated using the relation described in (16). Depending on the application, it may be beneficial to use the ℓ_1 -norm normalization [5] instead of the ℓ_2 -norm, which leads to the following alternative update rule for core tensor (referred to as the ℓ_1 HALS-NTD algorithm):

$$\underline{\mathbf{G}} \leftarrow \left[\underline{\mathbf{G}} \circledast \left(\underline{\mathbf{Y}} \oslash \widehat{\underline{\mathbf{Y}}} \right) \times \{\mathbf{A}^T\} \right]_+. \quad (17)$$

The core tensor can also be updated by the global ALS learning rule

$$\underline{\mathbf{G}} \leftarrow \left[\underline{\mathbf{Y}} \times \{\mathbf{A}^\dagger\} \right]_+ \quad (18)$$

where \mathbf{A}^\dagger denotes the Moore-Penrose pseudo-inverse of factor \mathbf{A} .

3 Experiments

The proposed algorithms were analyzed in 3 experiments involving applications of data denoising, classification. One experiment was analyzed using a synthetic benchmark, and two others were performed on real-world EEG datasets. The ℓ_2 HALS algorithm was emphasized, and its estimated core tensor is ready to identify the complex interactive relations among hidden components.

3.1 Noisy Data Reconstruction

We constructed a 3-rd order tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{200 \times 200 \times 200}$ corrupted by additive Gaussian noise SNR = -10 dB by benchmarks ACPos24sparse10 [8], and a random core tensor $\underline{\mathbf{G}} \in \mathbb{R}^{4 \times 5 \times 4}$. The noisy tensor was decomposed to retrieve the hidden factors and a core tensor. Then we built up reconstructed tensors. The ℓ_2 HALS algorithm was compared with the HONMF [4] algorithm, and also with HOOI algorithm for Tucker decomposition [9]. All algorithms were evaluated under the same condition of difference of FIT value (1e-6) using Peak Signal to Noise Ratio (PSNR) for all frontal slices. The results are illustrated in Fig. 2, with PSNR [dB] and FIT (%) values. The ℓ_2 HALS algorithm returned the most successful reconstruction with PSNR = 43.92 dB (Fig. 2(a)). Although the HOOI algorithm does not require nonnegative constraint, that algorithm gave a quite good result, but its reconstructed tensor was still noisy. The ℓ_2 HALS and HONMF algorithms converged after 133.04 and 249.13 seconds, respectively.

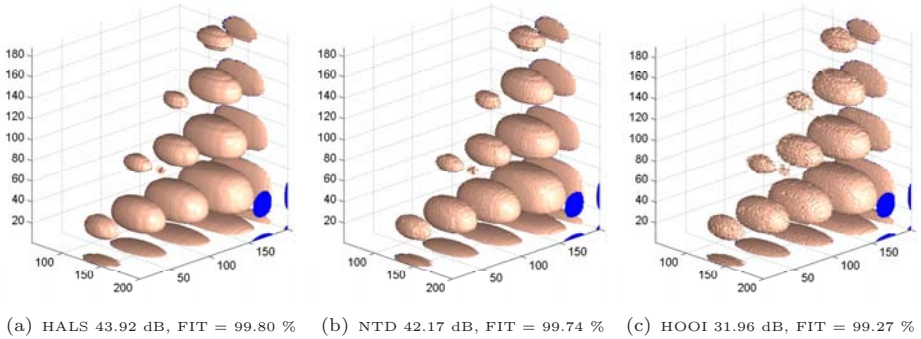


Fig. 2. Iso-surface visualization of simulation results for Example 1 with tensor $\underline{\mathbf{Y}} \in \mathbb{R}_+^{200 \times 200 \times 200}$ corrupted by Gaussian noise with SNR = -10dB

3.2 Classification of EEG Signal

We illustrate the HALS NTD algorithm with a simple example of EEG classification according to the nature of the stimulus for the benchmark `EEG_AV_stimuli` [10]: auditory stimulus, visual stimulus, both the auditory and the visual stimuli simultaneously. EEG signals were recorded from 61 channels during 1.5 seconds after stimulus presentation at a sampling rate of 1 kHz, and in 25 trials. The observed nonnegative tensor $\underline{\mathbf{Y}}$ consists the WTav measurements in the time-frequency domain using the complex Morlet wavelet: 61 channels \times 3906 frequency-time (31 frequency bins (10-40 Hz) \times 126 time frames (0-500ms)) \times 3 classes. The number of components was set to three, and the estimated factors and core tensor are illustrated in Fig. 3.

An advantage of the ℓ_2 HALS algorithm is that all the component vectors are unit length vectors, hence the coefficients of the core tensor $\underline{\mathbf{G}}$ express the energy of rank-one tensors built up from the basis components $\mathbf{a}_j^{(n)}$, ($n = 1, 2, \dots, N$, $j = 1, 2, \dots, J_n$), and is ready to evaluate the complex interactions between components using the Joint Rate (JR) index [11]. For example, the relation of spatial and category components can be identified using the JR index given in Fig. 3(e): the auditory class (component 2) interacts predominantly with the third spatial component, whereas the visual class (component 3) links with the second spatial component, and the auditory+visual class (component 1) links with all the spatial ones.

3.3 BCI Experiment

This example illustrates the analysis of real-world EEG data containing the evoked spectral perturbation (ERSP) measurements of EEG signals recorded from 62 electrodes during right and left hand motor imagery [12]. The observed tensor has size of 62 channels \times 25 frequency bins \times 1000 time frames \times 2 classes (Left/Right). The ℓ_2 HALS algorithm returned the decomposition results with the core tensor size of $4 \times 3 \times 3 \times 2$, and sparsity and orthogonality constraints

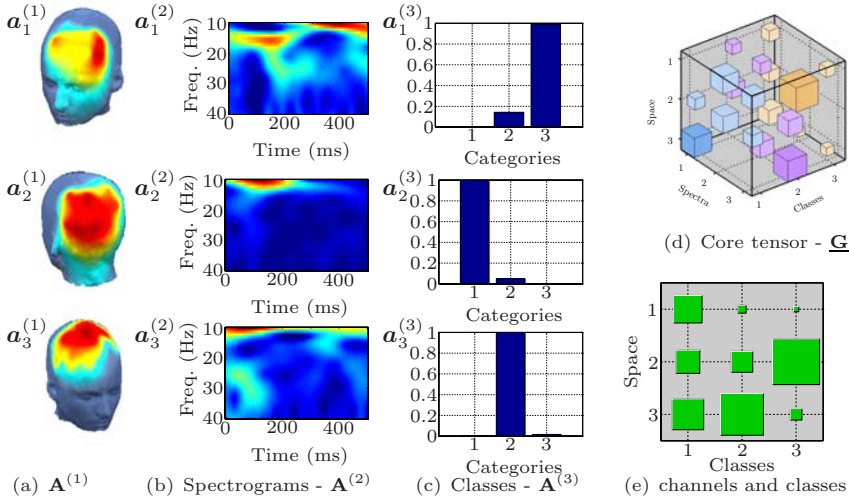


Fig. 3. Visualization of components for example 2 (a) spherically-spline EEG field maps; (b) spectral components expressed by factor $\mathbf{A}^{(2)}$; (c) category factor $\mathbf{A}^{(3)}$ for 3 classes: $\mathbf{a}_1^{(3)}$ - auditory-visual class, $\mathbf{a}_2^{(3)}$ - auditory class, and $\mathbf{a}_3^{(3)}$ - visual class; (d)-(e) Hinton diagrams of core tensor, and JR index of spatial and category components

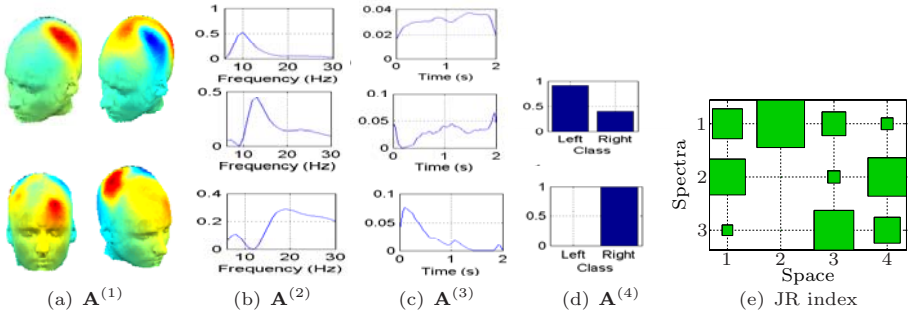


Fig. 4. Illustration for example 3: (a)-(d) factor visualizations; (e) JR index of interactive relations: spectral and spatial components

on factors $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ displayed in Fig. 4. The JR matrix indicating the interactive relation between the spectral components $\mathbf{A}^{(2)}$ and the spatial components $\mathbf{A}^{(1)}$ is displayed in Fig. 4(e). Component $\mathbf{a}_1^{(3)}$ corresponds to class-1 (right-hand imagery) with a larger amplitude on the left hemisphere, and lower amplitude for the right one. Whereas component $\mathbf{a}_4^{(1)}$ for class-2 (left-hand imagery) shows ERD on the left hemisphere and ERS on the right hemisphere (see Fig. 4(d)). Both these components are mainly affected by the spectral component $\mathbf{a}_2^{(2)}$ (see Fig. 4(e)).

4 Conclusion

We presented new local NTD algorithms, and confirmed their robustness to noise and good convergence properties in synthetic and real-world data sets. The proposed algorithms can resolve large scale problem due to sequentially update components instead of processing based on full data. The result core tensor of the ℓ_2 HALS algorithm is ready to evaluate complex interactive relations between components of factors using the Joint Rate index.

References

1. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.: Nonnegative Matrix and Tensor Factorizations. Wiley, Chichester (2009)
2. Tucker, L.: Some mathematical notes on three-mode factor analysis. *Psychometrika* 31, 279–311 (1966)
3. Kolda, T., Bader, B.: Tensor decompositions and applications. *SIAM Review* 51(3) (in print, September 2009)
4. Mørup, M., Hansen, L., Arnfred, S.: Algorithms for Sparse Nonnegative Tucker Decompositions. *Neural Computation* 20, 2112–2131 (2008)
5. Phan, A.H., Cichocki, A.: Fast and efficient algorithms for nonnegative Tucker decomposition. In: Sun, F., Zhang, J., Tan, Y., Cao, J., Yu, W. (eds.) *ISNN 2008, Part II*. LNCS, vol. 5264, pp. 772–782. Springer, Heidelberg (2008)
6. Cichocki, A., Phan, A.H.: Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE* (invited paper) (March 2009)
7. Kim, Y.D., Choi, S.: Nonnegative Tucker Decomposition. In: *Proc. of Conf. Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, Minnesota (June 2007)
8. Cichocki, A., Zdunek, R.: NTFLAB for Signal Processing. Technical report, Laboratory for Advanced Brain Signal Processing, BSI, RIKEN, Saitama, Japan (2006)
9. Bader, B., Kolda, T.: *MATLAB Tensor Toolbox Version 2.2* (January 2007)
10. Bakardjian, H., Cichocki, A.: Extraction and classification of common independent components in single-trial crossmodal cortical responses. In: *Proceedings of the 5th Annual Meeting of the International Multisensory Research Forum*, Barcelona, Spain, June 2004, pp. 26–27 (2004)
11. Phan, A.H., Cichocki, A.: Analysis of interactions among hidden components for Tucker model. In: *APSIPA Annual Summit and Conference* (2009)
12. Cichocki, A., Washizawa, Y., Rutkowski, T., Bakardjian, H., Phan, A.H., Choi, S., Lee, H., Zhao, Q., Zhang, L., Li, Y.: Noninvasive BCIs: Multiway signal-processing array decompositions. *Computer* 41(10), 34–42 (2008)

Comparing Large Datasets Structures through Unsupervised Learning^{*}

Guénaél Cabanes and Younès Bennani

LIPN-CNRS, UMR 7030, Université de Paris 13
99, Avenue J-B. Clément, 93430 Villetaneuse, France
cabanes@lipn.univ-paris13.fr

Abstract. In data mining, the problem of measuring similarities between different subsets is an important issue which has been little investigated up to now. In this paper, a novel method is proposed based on unsupervised learning. Different subsets of a dataset are characterized by means of a model which implicitly corresponds to a set of prototypes, each one capturing a different modality of the data. Then, structural differences between two subsets are reflected in the corresponding model. Differences between models are detected using a similarity measure based on data density. Experiments over synthetic and real datasets illustrate the effectiveness, efficiency, and insights provided by our approach.

1 Introduction

In recent years, the datasets' size has shown an exponential growth. Studies exhibit that the amount of data doubles every year. However, the ability to analyze these data remains inadequate. The problem of mining these data to measure similarities between different datasets becomes an important issue which has been little investigated up to now. A major application may be the analysis of time evolving datasets, by computing a model of the data structure over different periods of time, and comparing them to detect the changes when they occurred. Nevertheless, there are many other possible applications, like large datasets comparison, clustering merging, stability measure and so on.

As the study of data streams and large databases is a difficult problem because of the computing costs and the big storage volumes involved, two issues appear to play a key role in such an analysis: (i) a good condensed description of the data properties [1,2] and (ii) a measure capable of detecting changes in the data structure [3,4]. In this paper we propose a new algorithm which is able to perform these two tasks. The solution we propose consists of an algorithm, which first constructs an abstract representation of the datasets to compare and then evaluates the dissimilarity between them based on this representation. The abstract representation is based on the learning of a variant of Self-Organizing Map (SOM) [5], which is enriched with structural information extracted from the data. Then we propose a method to estimate, from the abstract representation, the underlying data density function. The dissimilarity is a measure of the divergence

^{*} This work was supported in part by the *CADI* project (N° ANR-07 TLOG 003) financed by the ANR (Agence Nationale de la Recherche).

between two estimated densities. A great advantage of this method is that each enriched SOM is at the same time a very informative and a highly condensed description of the data structure that can be stored easily for a future use. Also, as the algorithm is very effective both in terms of computational complexity and in terms of memory requirements, it can be used for comparing large datasets or for detecting structural changes in data-streams.

The remainder of this paper is organized as follows. Section 2 presents the new algorithm. Section 3 describes the validation protocol and some results. Conclusion and future work perspectives are given in Section 4.

2 A New Two-Levels Algorithm to Compare Data Structure

The basic assumption in this work is that data are described as vectors of numeric attributes and that the datasets to compare have the same type. First, each dataset is modeled using an enriched Self-organizing Map (SOM) model (adapted from [6]), constructing an abstract representation which is supposed to capture the essential data structure. Then, each dataset density function is estimated from the abstract representation. Finally, different datasets are compared using a dissimilarity measure based upon the density functions.

The idea is to combine the dimension reduction and the fast learning SOM capabilities in the first level to construct a new reduced vector space, then applies other analysis in this new space. These are called two-levels methods. The two-levels methods are known to reduce greatly the computational time, the effects of noise and the “curse of dimensionality” [6]. Furthermore, it allows some visual interpretation of the result using the two-dimensional map generated by the SOM.

2.1 Abstract Algorithm Schema

The algorithm proceeds in three steps :

1. The first step is the learning of the enriched SOM. During the learning, each SOM prototype is extended with novel information extracted from the data. These structural informations will be used in the second step to infer the density function. More specifically, the attributes added to each prototype are:
 - *Density modes*. It is a measure of the data density surrounding the prototype (local density). The local density is a measure of the amount of data present in an area of the input space. We use a Gaussian kernel estimator [7] for this task.
 - *Local variability*. It is a measure of the data variability that is represented by the prototype. It can be defined as the average distance between the prototypes and the represented data.
 - *The neighborhood*. This is a prototype’s neighborhood measure. The neighborhood value of two prototypes is the number of data that are well represented by each one.
2. The second step is the construction, from each enriched SOM, of a density function which will be used to estimate the density of the input space. This function is constructed by induction from the information associated to the prototypes of the SOM, and is represented as a mixture model of spherical normal functions.

3. The last step accomplishes the comparison of two different datasets, using a dissimilarity measure able to compare the two density functions constructed in the previous steps.

2.2 Prototypes Enrichment

In this step some global information is extracted from the data and stored in the prototypes during the learning of the SOM. The Kohonen SOM can be classified as a competitive unsupervised learning neural network [5]. A SOM consists in a two dimensional map of M neurons (units) which are connected to n inputs according to n weights connections $w_j = (w_{1j}, \dots, w_{nj})$ (also called prototypes) and to their neighbors with topological links. The training set is used to organize these maps under topological constraints of the input space. Thus, an optimal spatial organization is determined by the SOM from the input data.

In our algorithm, the SOM's prototypes will be "enriched" by adding new numerical values extracted from the dataset. The enrichment algorithm proceeds in three phases:

Input :

- The data $X = \{x_k\}_{k=1}^N$.

Output :

- The density D_i and the local variability s_i associated to each prototype w_i .
- The neighborhood values $v_{i,j}$ associated with each pair of prototype w_i and w_j .

Algorithm:

1. Initialization :

- Initialize the SOM parameters
- $\forall i, j$ initialize to zero the local densities (D_i), the neighborhood values ($v_{i,j}$), the local variability (s_i) and the number of data represented by w_i (N_i).

2. Choose randomly a data $x_k \in X$:

- Compute $d(w, x_k)$, the euclidean distance between the data x_k and each prototype w_i .
- Find the two closest prototypes (BMUs: Best Match Units) w_{u^*} and $w_{u^{**}}$:

$$u^* = \arg \min_i (d(w_i, x_k)) \quad \text{and} \quad u^{**} = \arg \min_{i \neq u^*} (d(w_i, x_k))$$

3. Update structural values :

- Number of data: $N_{u^*} = N_{u^*} + 1$.
- Variability: $s_{u^*} = s_{u^*} + d(w_{u^*}, x_k)$.
- Density: $\forall i, D_i = D_i + \frac{1}{\sqrt{2\pi}h} e^{-\frac{d(w_i, x_k)^2}{2h^2}}$.
- Neighborhood: $v_{u^*, u^{**}} = v_{u^*, u^{**}} + 1$.

4. **Update the SOM prototypes** w_i as defined in [5].
5. **repeat T times step 2 to 4.**
6. **Final structural values:** $\forall i, s_i = s_i/N_i$ and $D_i = D_i/N$.

In this study we used the default parameters of the SOM Toolbox [8] for the learning of the SOM and we use $T = \max(N, 50 \times M)$ as in [8]. The number M of prototypes must neither be too small (the SOM does not fit the data well) nor too large (time consuming). To choose M close to \sqrt{N} seems to be a good trade-off [8]. The last parameter to choose is the bandwidth h . The choice of h is important for good results, but its optimal value is difficult to calculate and time consuming (see [9]). A heuristic that seems relevant and gives good results consists in defining h as the average distance between a prototype and its closest neighbor [6].

At the end of this process, each prototype is associated with a density and a variability value, and each pair of prototypes is associated with a neighborhood value. The substantial information about the structure of the data is captured by these values. Then, it is no longer necessary to keep data in memory.

2.3 Estimation of the Density Function

The objective of this step is to estimate the density function which associates a density value to each point of the input space. We already have an estimation of the value of this function at the position of the prototypes (i.e. D_i). We must infer from this an approximation of the function.

Our hypothesis here is that this function may be properly approximated in the form of a mixture of Gaussian kernels. Each kernel K is a Gaussian function centered on a prototype. The density function can therefore be written as:

$$f(x) = \sum_{i=1}^M \alpha_i K_i(x) \quad \text{with} \quad K_i(x) = \frac{1}{N\sqrt{2\pi}h_i} e^{-\frac{d(w_i, x)^2}{2h_i^2}}$$

The most popular method to fit mixture models (i.e. to find h_i and α_i) is the expectation-maximization (EM) algorithm [10]. However, this algorithm needs to work in the data input space. As here we work on enriched SOM instead of dataset, we can't use EM algorithm.

Thus, we propose the heuristic to choose h_i :

$$h_i = \frac{\sum_j \frac{v_{i,j}}{N_i + N_j} (s_i N_i + d_{i,j} N_j)}{\sum_j v_{i,j}}$$

$d_{i,j}$ is the euclidean distance between w_i and w_j . The idea is that h_i is the standard deviation of data represented by K_i . These data are also represented by w_i and their neighbors. Then h_i depends on the variability s_i computed for w_i and the distance $d_{i,j}$ between w_i and his neighbors, weighted by the number of data represented by each prototype and the connectivity value between w_i and his neighborhood.

Now, since the density D for each prototype w is known ($f(w_i) = D_i$), we can use a gradient descent method to determine the weights α_i . The α_i are initialized with the

values of D_i , then these values are reduced gradually to better fit $D = \sum_{i=1}^M \alpha_i K_i(w)$. To do this, we optimize the following criterion:

$$\alpha = \arg \min_{\alpha} \frac{1}{M} \sum_{i=1}^M \left[\sum_{j=1}^M (\alpha_j K_j(w_i)) - D_i \right]^2$$

Thus, we now have a density function that is a model of the dataset represented by the enriched SOM.

2.4 Algorithm Complexity

The complexity of the algorithm is scaled as $O(T \times M)$, with T the number of steps and M the number of prototypes in the SOM. It is recommended to set at least $T > 10 \times M$ for a good convergence of the SOM [5]. In this study we use $T = \max(N, 50 \times M)$ as in [8]. This means that if $N > 50 \times M$ (large database), the complexity of the algorithm is $O(N \times M)$, i.e. is linear in N for a fixed size of the SOM. Then the whole process is very fast and is suited for the treatment of large databases. Also very large databases can be handled by fixing $T < N$ (this is similar as working on a random subsample of the database).

This is much faster than traditional density estimator algorithms as the Kernel estimator [7] (that also needs to keep all data in memory) or the Gaussian Mixture Model [11] estimated with the EM algorithm (as the convergence speed can become extraordinarily slow [12,13]).

2.5 The Dissimilarity Measure

We can now define a measure of dissimilarity between two datasets A and B , represented by two SOMs: $SOM_A = [\{w_i^A\}_{i=1}^{M^A}, f^A]$ and $SOM_B = [\{w_i^B\}_{i=1}^{M^B}, f^B]$. With M^A and M^B the number of prototypes in models A and B , and f^A and f^B the density function of A and B computed in §2.3.

The dissimilarity between A and B is given by:

$$CBd(A, B) = \frac{\sum_{i=1}^{M^A} f^A(w_i^A) \log \left(\frac{f^A(w_i^A)}{f^B(w_i^A)} \right)}{M^A} + \frac{\sum_{j=1}^{M^B} f^B(w_j^B) \log \left(\frac{f^B(w_j^B)}{f^A(w_j^B)} \right)}{M^B}$$

The idea is to compare the density functions f^A and f^B for each prototype w of A and B . If the distributions are identical, these two values must be very close. This measure is an adaptation of the weighted Monte Carlo approximation of the symmetrical Kullback–Leibler measure (see [14]), using the prototypes of a SOM as a sample of the database.

3 Validation

3.1 Description of the Used Datasets

In order to demonstrate the performance of the proposed dissimilarity measure, we used nine artificial datasets generators and one real dataset.

“Ring 1”, “Ring 2”, “Ring 3”, “Spiral 1” and “Spiral 2” are 5 two-dimensional non-convex distributions with different density and variance. “Ring 1” is a ring of radius 1 (high density), “Ring 2” a ring of radius 3 (low density) and “Ring 3” a ring of radius 5 (average density); “Spiral 1” and “Spiral 2” are two parallel spirals. The density in the spirals decreases with the radius. Datasets from these distributions can be generated randomly.

“Noise 1” to “Noise 4” are 4 different two-dimensional distributions, composed of different Gaussian distributions and a heavy homogeneous noise.

Finally, the “Shuttle” real dataset come from the UCI repository. It’s a nine-dimensional dataset with 58000 instances. These data are divided in seven class. Approximately 80% of the data belongs to class 1.

3.2 Validity of the Dissimilarity Measure

It’s impossible to prove that two distributions are exactly the same. Anyway, a low dissimilarity value is only consistent with a similar distribution, and does of course give an indication of the similarity between the two sample distributions. On the other hand, a very high dissimilarity does show, to the given level of significance, that the distributions are different. Then, if our measure of dissimilarity is efficient, it should be possible to compare different datasets (with the same attributes) to detect the presence of similar distributions, i.e. the dissimilarity of datasets generated from the same distribution law must be much smaller than the dissimilarity of datasets generated from very different distribution.

To test this hypothesis we applied the following protocol:

1. We generated 250 different datasets from the “Ring 1”, “Ring 2”, “Ring 3”, “Spiral 1” and “Spiral 2” distributions (50 datasets from each). Each sets contain between 500 and 50000 data.
2. For each of these datasets, we learned an enriched SOM to obtain a set of representative prototypes of the data. The number of prototypes is randomly chosen from 50 to 500.
3. We computed a density function for each SOM and compared them to each other with the proposed dissimilarity measure.
4. Each SOM was labeled depending on the distribution represented (labels are “ring 1”, “ring 2”, “ring 3”, “spiral 1” and “spiral 2”). We then calculated an index of compactness and separability of SOM having the same label with the generalized index of Dunn [15]. This index is even bigger than SOM with the same label are similar together and dissimilar to other labels, according to the dissimilarity function used.

We used the same protocol with “Noise 1” to “Noise 4” and with the “Shuttle” database. Two kinds of distributions have been extracted from the “Shuttle” database, by using random sub-sampling (with replacement) of data from class 1 (“Shuttle 1”) and data from other classes (“Shuttle 2”).

We compared the results with some distance-based measures usually used to compare two sets of data (here we compare two sets of prototypes from the SOMs). These

measures are the average distance (Ad: the average distance between all pair of prototypes in the two SOMs), the minimum distance (Md: the smallest Euclidean distance between prototypes in the two SOMs) and the Ward distance (Wd: The distance between the two centroids, with some weight depending on the number of prototypes in the two SOMs)) [16]. All results are in Table I. The visual inspection of the dissimilarity matrix obtained from the different measures is consistent with the values of the Dunn index (Table I) to show that the density-based proposed similarity measure (CBd) is much more effective than distance based measures (see Fig. 1 for an example).

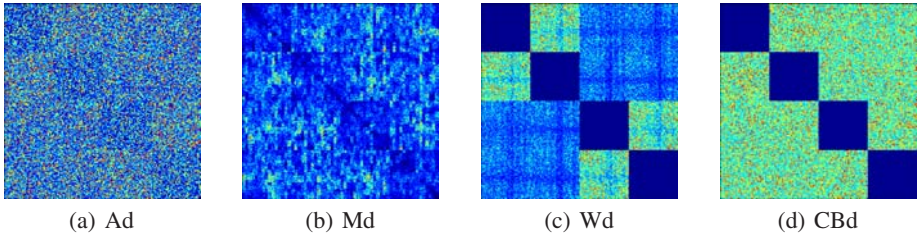


Fig. 1. Visualizations of the dissimilarity matrix obtained from the different measures for the comparisons of the different “Noise” 1 to 4 distributions. Darker cells mean higher similarity between the two distributions. Datasets from the same distribution are sorted, then their comparisons appear as four box along the diagonal of the matrix.

As shown in table I our dissimilarity measure using density is much more effective than measures that only use the distances between prototypes. Indeed, the Dunn index for density based measure is much higher than distance based ones for the three kinds of datasets tested: non-convex data, noisy data and real data. This means that our measure of dissimilarity is much more effective for the detection of similar dataset.

Table 1. Value of the Dunn index obtained from various dissimilarity measures to compare various data distributions

Distributions to compare	Average	Minimum	Ward	Proposed
Ring 1 to 3 + Spiral 1 and 2	0.4	0.9	0.5	1.6
Noise 1 to 4	1.1	1.4	22.0	115.3
Shuffle 1 and 2	1.1	16.5	6.3	27.6

4 Conclusion and Future Works

In this article, we proposed a new algorithm for modeling data structure, based on the learning of a SOM, and a measure of dissimilarity between cluster structures. The advantages of this algorithm are not only the low computational cost and the low memory requirement, but also the high accuracy achieved in fitting the structure of the modeled datasets. These properties make it possible to apply the algorithm to the analysis

of large data bases, and especially large data streams, which requires both speed and economy of resources. The results obtained on the basis of artificial and real datasets are very encouraging.

The continuation of our work will focus on the validation of the algorithm on more real case studies. More specifically, it will be applied to the analysis of real data streams, commonly used as benchmark for this class of algorithms. We will also try to extend the method to structured or symbolic datasets, via kernel-based SOM algorithm. Finally, we wish to deepen the concept of stability applied to this kind of analysis.

References

1. Gehrke, J., Korn, F., Srivastava, D.: On computing correlated aggregates over continual data streams. In: SIGMOD Conference (2001)
2. Manku, G.S., Motwani, R.: Approximate frequency counts over data streams. In: VLDB, pp. 346–357 (2002)
3. Cao, F., Ester, M., Qian, W., Zhou, A.: Density-based clustering over an evolving data stream with noise. In: 2006 SIAM Conference on Data Mining (2006)
4. Aggarwal, C., Yu, P.: A Survey of Synopsis Construction Methods in Data Streams. In: Aggarwal, C. (ed.) Data Streams: Models and Algorithms. Springer, Heidelberg (2007)
5. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (2001)
6. Cabanes, G., Bennani, Y.: A local density-based simultaneous two-level algorithm for topographic clustering. In: IJCNN, pp. 1176–1182. IEEE, Los Alamitos (2008)
7. Silverman, B.: Using kernel density estimates to investigate multi-modality. *Journal of the Royal Statistical Society, Series B* 43, 97–99 (1981)
8. Vesanto, J.: Neural network tool for data mining: SOM Toolbox (2000)
9. Sain, S., Baggerly, K., Scott, D.: Cross-Validation of Multivariate Densities. *Journal of the American Statistical Association* 89, 807–817 (1994)
10. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* 39, 1–38 (1977)
11. Fraley, C., Raftery, A.E.: Model-based clustering, discriminant analysis, and density estimation. *J. Am. Stat. Assoc.* 97(458), 611–631 (2002)
12. Redner, R.A., Walker, H.F.: Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review* 26(2), 195–239 (1984)
13. Park, H., Ozeki, T.: Singularity and Slow Convergence of the EM algorithm for Gaussian Mixtures. *Neural Process Letters* 29, 45–59 (2009)
14. Hershey, J.R., Olsen, P.A.: Approximating the kullback leibler divergence between gaussian mixture models. In: IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 4, pp. 317–320 (2007)
15. Dunn, J.C.: Well separated clusters and optimal fuzzy partitions. *J. Cybern.* 4, 95–104 (1974)
16. Jain, A.K., Dubes, R.C.: Algorithms for clustering data. Prentice-Hall, Inc, Upper Saddle River (1988)

Applying Duo Output Neural Networks to Solve Single Output Regression Problem

Pawalai Kraipeerapun¹, Somkid Amornsamankul²,
Chun Che Fung³, and Sathit Nakkrasae¹

¹ Department of Computer Science, Faculty of Science, Ramkhamhaeng University,
Bangkok, Thailand

pawalai@ru.ac.th, sathit@ru.ac.th

² Department of Mathematics, Faculty of Science, Mahidol University, Thailand
Centre of Excellence in Mathematics, CHE, Sriyudhaya Rd., Bangkok, Thailand

scsam@mahidol.ac.th

³ School of Information Technology, Murdoch University, Perth, Australia
l.fung@murdoch.edu.au

Abstract. This paper proposes a novel approach to solve a single output regression problem using duo output neural network. A pair of duo output neural networks is created. The first neural network is trained to provide two outputs which are the truth and the falsity values. The second neural network is also trained to provide two outputs; however, the sequence of the outputs is organized in reverse order of the first one. Therefore, the two outputs of this neural network is the falsity and the truth values. All the truth and the non-falsity values obtained from both neural networks are then averaged to give the final output. We experiment our proposed approach to the classical benchmark problems which are housing, concrete compressive strength, and computer hardware data sets from the UCI machine learning repository. It is found that the proposed approach provides better performance when compared to the complementary neural networks, backpropagation neural networks, and support vector regression with linear, polynomial, and radial basis function kernels.

1 Introduction

Neural network is one of the most popular predictors used to solve the regression problem. One of the reasons is that the neural network can provide better performance when compare to the statistical methods [1-3] and the support vector regression [4-6].

In recent years, complementary neural networks (CMTNN) have been used to solve both classification and regression problems [7, 8]. Instead of considering only the truth output obtained from neural networks, complementary neural networks consider both truth and falsity outputs predicted from the truth and falsity neural networks, respectively. Both neural networks have the same architecture and apply the same parameter values for training. However, the target

output used to train the falsity neural network is the complement of the target output used to train the truth neural network. Several aggregation techniques have been proposed based on CMTNN. One of those aggregations is the equal weight averaging which is the simple averaging between the truth and non-falsity outputs obtained from the truth and falsity neural networks. The result obtained from these aggregation techniques have been found to provide better accuracy results compared to the traditional neural network trained with only the truth target output [7, 8]. In this paper, we proposed a duo output neural network (DONN) based on complementary neural networks. Instead of applying a pair of neural networks, a single neural network with two outputs is utilized. In order to get better results, a pair of neural networks with two outputs is considered. We experiment our proposed approach to the classical benchmark problems including housing, concrete compressive strength [9], and computer hardware from the UCI machine learning repository [10].

The rest of this paper is organized as follows. Section 2 explains the duo output neural network and the proposed aggregation technique used for single output regression. Section 3 describes the data set and results of our experiments. Conclusions and future work are presented in Section 4.

2 Duo Output Neural Network (DONN)

Duo output neural network is a neural network trained with two target outputs, which are complement to each other. Let $T_{target}(x_i)$ be the true target output for the input pattern $x_i, i = 1, 2, 3, \dots, n$ where n is the total number of training input patterns. Let $F_{target}(x_i)$ be the false target output for the input pattern x_i . The false target output is the complement of the true target output. Hence, the false target output can be computed as follows.

$$F_{target}(x_i) = 1 - T_{target}(x_i) \quad (1)$$

The duo output neural network is trained using two complementary target values (T_{target} and F_{target}) in order to predict two complementary outputs which are the truth and falsity output values. In this case, the sequence of the outputs can be organized in two ways: truth-falsity and falsity-truth. Therefore, two duo output neural networks are created to support both types of output. Fig. 1 shows our proposed duo output neural network model in the training phase. Two neural networks are trained. The first neural network, NN_1 , is trained to predict the truth output T_{train_1} and the falsity output F_{train_1} using the true target and the false target values, respectively. On the other hand, the second neural network, NN_2 , is trained to predict the falsity output F_{train_2} and the truth output T_{train_2} using the false target and the true target values, respectively. The ensemble of these two duo output neural networks can provide us better accuracy results when dealing with the unknown input data.

In the testing phase, each unknown input pattern y_j is assigned to the two duo output neural networks where $j = 1, 2, 3, \dots, t$ and t is the total number of unknown input patterns. Fig. 2 shows the proposed duo output neural network

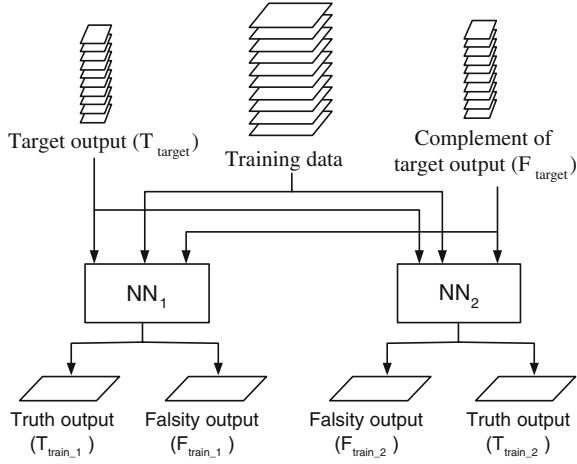


Fig. 1. Duo output neural network model (Training Phase)

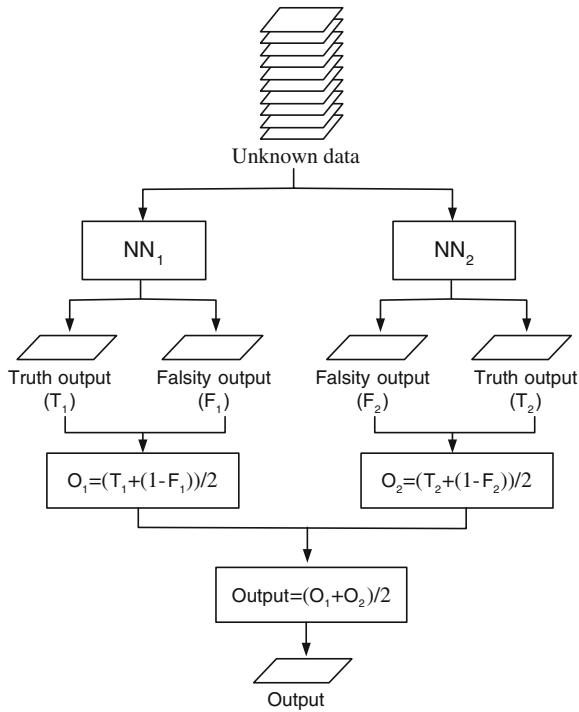


Fig. 2. Duo output neural networks model (Testing Phase)

model in the testing phase. Let $T_1(y_j)$ and $F_1(y_j)$ be the truth and the falsity outputs for the unknown input pattern y_j of the first neural network (NN_1). The truth and the non-falsity outputs can be aggregated to provide the combined output as shown in the equation below.

$$O_1(y_j) = \frac{T_1(y_j) + (1 - F_1(y_j))}{2} \quad (2)$$

Let $F_2(y_j)$ and $T_2(y_j)$ be the falsity and the truth outputs for the unknown input pattern y_j of the second neural network (NN_2). The combined output of the truth and the non-falsity outputs can be computed as follows.

$$O_2(y_j) = \frac{T_2(y_j) + (1 - F_2(y_j))}{2} \quad (3)$$

The regression output for each unknown input pattern y_j can be computed as the average between both outputs $O_1(y_j)$ and $O_2(y_j)$ as shown below.

$$O(y_j) = \frac{O_1(y_j) + O_2(y_j)}{2} \quad (4)$$

3 Experiments

3.1 Data Set

Three UCI data sets are used in this experiment. The characteristics of these data sets which are housing, concrete compressive strength, and computer hardware are shown in Table [1](#).

Table 1. UCI data sets used in this study

Name	Feature type	No. of features	No. of samples
Housing	numeric	13	506
Concrete	numeric	8	1030
Hardware	numeric	6	209

3.2 Experimental Methodology and Results

In this experiment, ten-fold cross validation is applied to each data set. For each fold, two feed-forward backpropagation neural networks are trained with the same parameter values except the initial weight. Both network apply different initial weights in order to increase diversity in the ensemble of the two networks. The first neural network is trained to predict the truth-falsity output values whereas the second neural network is trained to predict the falsity-truth output values. In both neural networks, the number of input-nodes is equal to the number of input features, which is 13, 8, and 6 for housing, concrete, and

Table 2. The comparison among the average of mean square error, MSE, (ten folds) obtained from SVR, BPNN, CMTNN, and DONN for housing, concrete, and hardware data sets

Technique	Mean Square Error (MSE)		
	Housing	Concrete	Hardware
SVR (linear)	0.045140	0.048829	0.017808
SVR (Polynomial)	0.040929	0.042365	0.018324
SVR (RBF)	0.049051	0.041987	0.019221
BPNN	0.030113	0.021595	0.005601
CMTNN (Equal weight averaging)	0.019275	0.017384	0.004377
DONN (Truth-Falsity)	0.020756	0.021249	0.006930
DONN (Falsity-Truth)	0.022923	0.013848	0.002899
DONN (Combination)	0.012304	0.014642	0.003767

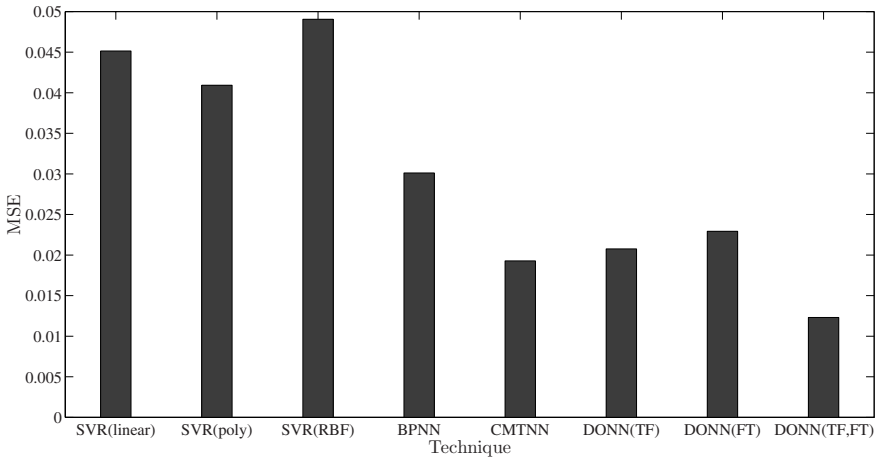


Fig. 3. The average of mean square error obtained from the test set of housing data

hardware data sets, respectively. They have one hidden layer constituting of $2n$ neurons where n is the number of input features. Hence, the number of neuron in the hidden layer for those data sets are 26, 16, and 12, respectively.

Table 2 shows the average of mean square error (MSE) obtained from the proposed duo output neural networks (DONN) compared to other existing estimators, which are backpropagation neural network (BPNN), support vector regression (SVR) with linear, polynomial, and radial basis function (RBF) kernels, as well as the complementary neural networks (CMTNN). These estimators are also experimented with ten-fold cross validation method. This table shows that our proposed duo output neural network based on the combination of truth-falsity and falsity-truth outputs yields better performance than other

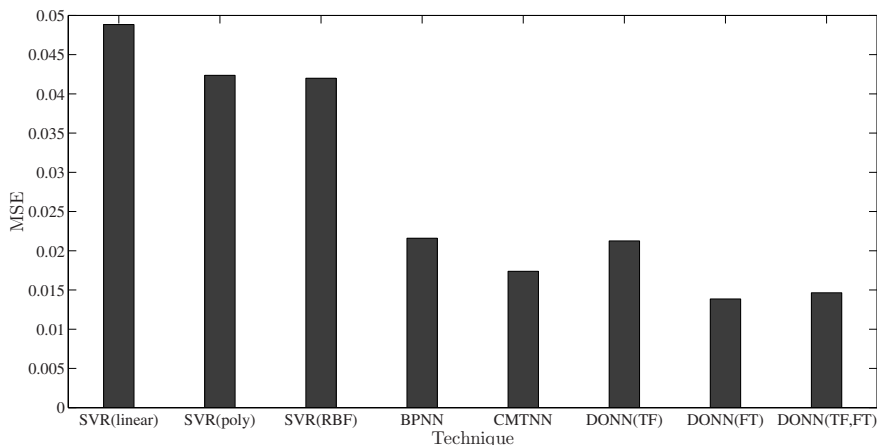


Fig. 4. The average of mean square error obtained from the test set of concrete data

Table 3. The average of percent improvement of the proposed DONN with the combination technique compared to the traditional SVR, BPNN, and CMTNN (ten folds)

Technique	DONN (Combination)		
	Housing	Concrete	Hardware
SVR(linear)	72.74%	70.01%	78.85%
SVR(polynomial)	69.94%	65.44%	79.44%
SVR(RBF)	74.92%	65.13%	80.40%
BPNN	59.14%	32.20%	32.75%
CMTNN (Equal weight averaging)	36.16%	15.77%	13.94%

techniques. Fig. 3, 4, and 5 show the graphical representation of the comparison of the mean square error among our technique and other techniques for the test set of housing, concrete, and hardware, respectively. From these figures, it can be observed that the technique based on neural networks provides better accuracy than the support vector regression. It is also found that the complementary neural networks and the combination of two opposite duo output neural networks outperform the traditional backpropagation neural networks. Moreover, it can be seen that the individual truth-falsity output neural network and the falsity-truth neural network may not provide good results when compared to CMTNN and BPNN. However, the combination between these two duo output neural networks yield better accuracy than other techniques. Table 3 shows the average of percent improvement of the proposed technique of the combination of two opposite duo output neural network compared to the traditional BPNN, SVR, and CMTNN.

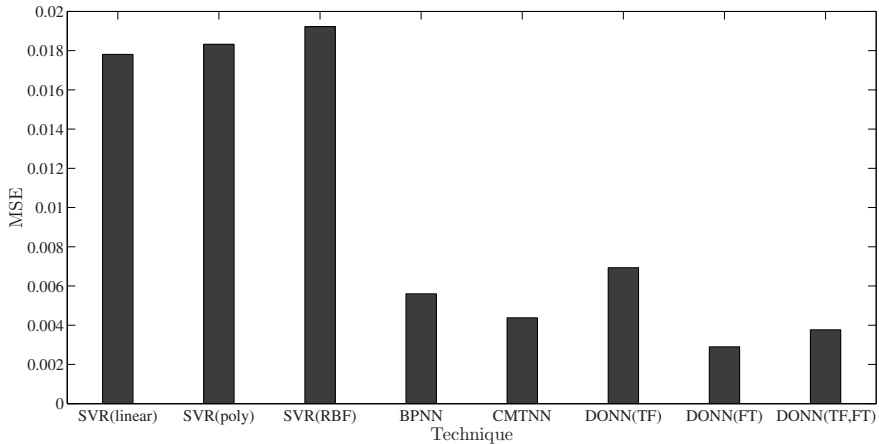


Fig. 5. The average of mean square error obtained from the test set of hardware data

4 Conclusion and Future Work

This paper applies the combination of two duo output neural networks to solve a single output regression problem. The first neural network provides the truth-falsity output whereas the second neural network provides the falsity-truth output. We found that the output obtained from the combination between both neural networks provide more accurate result than individual duo output neural network and other existing techniques. In the future, we will apply duo output neural network to solve the classification problem.

References

1. Paliwal, M., Kumar, U.A.: Neural networks and statistical techniques: A review of applications. *Expert Systems with Applications* 36, 2–17 (2009)
2. Crone, S.F., Lessmann, S., Pietsch, S.: Forecasting with Computational Intelligence - An Evaluation of Support Vector Regression and Artificial Neural Networks for Time Series Prediction. In: *Proceedings of International Joint Conference on Neural Networks*, Canada, July 2006, pp. 3159–3166 (2006)
3. Chen, W.-H., Shih, J.-Y.: Comparison of support-vector machines and back propagation neural networks in forecasting the six major Asian stock markets. *International Journal of Electronic Finance* 1(1), 49–67 (2006)
4. Osowski, S., Siwek, K., Markiewicz, T.: MLP and SVM Networks – a Comparative Study. In: *Proceedings of the 6th Nordic Signal Processing Symposium*, June 2004, pp. 37–40 (2004)
5. Meyer, D., Leisch, F., Hornik, K.: The support vector machine under test. *Neurocomputing* 55, 169–186 (2003)
6. Msiza, I.S., Nelwamondo, F.V., Marwala, T.: Water Demand Prediction using Artificial Neural Networks and Support Vector Regression. *Journal of Computers* 3(11), 1–8 (2008)

7. Kraipeerapun, P., Fung, C.C., Wong, K.W.: Ensemble Neural Networks Using Interval Neutrosophic Sets and Bagging. In: Proceedings of the Third International Conference on Natural Computation (ICNC 2007), Haikou, China, August 2007, vol. 1, pp. 386–390 (2007)
8. Kraipeerapun, P., Fung, C.C., Nakkrasae, S., Amornsamankul, S.: Applying Complementary Neural Networks to Porosity Prediction in Well Log Data Analysis. In: Proceedings of the 6th International Joint Conference on Computer Science and Software Engineering, Phuket, Thailand, May 13-15, 2009, pp. 319–323 (2009)
9. Yeh, I.-C.: Modeling of strength of high performance concrete using artificial neural networks. *Cement and Concrete Research* 28(12), 1797–1808 (1998)
10. Asuncion, A., Newman, D.: UCI machine learning repository (2007), <http://www.ics.uci.edu/~mllearn/MLRepository.html>

An Incremental Learning Algorithm for Resource Allocating Networks Based on Local Linear Regression*

Seiichi Ozawa and Keisuke Okamoto

Graduate School of Engineering, Kobe University, Kobe 657-8501, Japan
ozawasei@kobe-u.ac.jp

Abstract. To learn things incrementally without the catastrophic interference, we have proposed Resource Allocating Network with Long-Term Memory (RAN-LTM). In RAN-LTM, not only training data but also memory items stored in long-term memory are trained. In this paper, we propose an extended RAN-LTM called Resource Allocating Network by Local Linear Regression (RAN-LLR), in which its centers are not trained but selected based on output errors and the connections are updated by solving a linear regression problem. To reduce the computation and memory costs, the modified connections are restricted based on RBF activity. In the experiments, we first apply RAN-LLR to a one-dimensional function approximation problem to see how the negative interference is effectively suppressed. Then, the performance of RAN-LLR is evaluated for a real-world prediction problem. The experimental results demonstrate that the proposed RAN-LLR can learn fast and accurately with less memory costs compared with the conventional models.

1 Introduction

Memory-based learning such as Locally Weighted Regression (LWR) [9] and Radial-Basis Function (RBF) networks [2,3] is one of the most promising strategy in incremental learning. In this approach, (almost) all training samples are accumulated in a memory and they are utilized for predicting and learning whenever new data are given. However, when data consist of a large number of attributes (or features), the computation and memory costs could be serious especially under life-long learning environments.

For the memory-based RBF networks, keeping all the training data as the RBF centers could be unrealistic for high-dimensional data; therefore, an appropriate number of RBF centers should be selected from incoming training data. On the other hand, one of the difficulties in incremental learning is to suppress the so-called *catastrophic interference* which leads to unexpected forgetting of input-output relationships acquired in the past. This interference is mainly caused

* The authors would like to thank Professor Shigeo Abe for his helpful comments and discussions.

by modifying connection weights and RBF centers by which the input-output relations are represented in a distributed way.

To avoid the catastrophic interference effectively in RBF networks, there have been proposed several approaches [1]. A promising approach is that some representative input-output pairs are extracted from sequentially given training samples and some of them are trained with a current training sample. For this approach, an extended version of RBF networks called *Resource Allocating Network with Long-Term Memory* (RAN-LTM) and the two learning algorithms based on gradient descent algorithm [6] and linear regression method [5,7] have been proposed. In the former algorithm, not only the connections but also the centers and widths of RBFs can be learned. However, the learning may get stucked into local minima and its convergence is usually slow. In the latter method, optimal connections are always obtained by solving a set of linear equations in the least squared error sense. However, the computation and memory costs grow exponentially as the number of RBFs increases [7].

In this paper, to alleviate the increase in the computation and memory costs, we propose a new incremental learning algorithm for RAN-LTM in which the connections to be learned are restricted based on RBF activity. Since an RBF has local response to an input domain, the number of modified connections is roughly kept constant even if the input domain is growing over time.

2 Proposed Incremental Learning Algorithm

In this section, we propose a new RAN-LTM model which can reduce computation and memory costs by restricting modified connections. Let us call this model *Resource Allocating Network by Local Linear Regression* (RAN-LLR)D. The learning of RAN-LLR is divided as follows: (1) incremental allocation of RBFs, (2) creation & retrieval of memory items, (3) creation of pseudo training data, (3) selection of modified connections, and (4) learning of connections.

2.1 Incremental Allocation of RBFs

In the Platt's RAN, an RBF is added only when the distance $\|\mathbf{x}_p - \mathbf{c}^*\|$ between the p th input \mathbf{x}_p and the closest center \mathbf{c}^* is large and the error $E = \|\mathbf{T}_p - \mathbf{z}(\mathbf{x}_p)\|$ between the output $\mathbf{z}(\mathbf{x}_p)$ and the target \mathbf{T}_p is large. That is, when \mathbf{x}_p exists in an unknown region, an RBF is created by setting \mathbf{x}_p to its center.

In the linear regression approach, once RBFs are created, the centers are fixed afterwards. Therefore, the allocation of RBF centers can be affected by sequences of training data and it may result in non-optimal allocation in approximating a target function. In the proposed method, to keep the approximation error low, the necessity of allocating an RBF is checked every after the learning of connections. That is, after updating connections, the output error is recalculated; then, an RBF is added if the error is larger than a threshold.

2.2 Creation and Retrieval of Memory Items

As mentioned in [2.1](#), RBFs are added only when it is necessary for a network to maintain good approximation accuracy. Let J and K be the numbers of RBFs and outputs, respectively. In creating a new RBF, first the number of RBFs is incremented (i.e., $J \leftarrow J + 1$), then the center \mathbf{c}_J is set to the input \mathbf{x}_p (i.e., $\mathbf{c}_J = \mathbf{x}_p$). And the connection weights $\mathbf{W} \in R^{J \times K}$ are obtained such that the error between the target $\mathbf{T}_p \in R^K$ and the output $\mathbf{z}(\mathbf{x}_p)$ is reduced to zero. This can be achieved by updating \mathbf{W} as follows:

$$\mathbf{W}^{NEW} = \begin{bmatrix} \mathbf{W}^{OLD} \\ \mathbf{T}_p - \mathbf{z}(\mathbf{x}_p) \end{bmatrix} \tag{1}$$

After creating a new RBF, the training data $(\mathbf{x}_p, \mathbf{T}_p)$ is stored in the long-term memory (LTM) as a *memory item* because it represents crucial input-output relation which should not be forgotten over the future learning stages. Let \mathbf{M}_J be the J th memory item which is created when the J th RBF is allocated. Note that J is the number of memory items as well as that of RBFs.

In the proposed method, the memory items that are retrieved from LTM are learned with training data. From the computational point of view, they should be restricted to essential ones that can suppress the interference. Considering that the interference mainly occurs at the connections to active RBFs whose outputs y_j are larger than a threshold η_3 , the input domain supported by the active RBFs is vulnerable to the interference. Therefore, the memory items associated with the active RBFs should be retrieved and learned. Let us define an index set S_A of active RBFs: $S_A = \{j \mid y_j \geq \eta_3; j = 1 \cdots J\}$. Then, the memory items to be retrieved are represented by $\mathbf{M}_j = (\mathbf{c}_j, \mathbf{T}_j)$ ($j \in S_A$).

2.3 Creation of Pseudo Training Data

Let $|S_A|$ be the number of active RBFs when a training data \mathbf{x}_p is given. Then, the total number of training data and retrieved memory items is $|S_A| + 1$, while the number of connection weights is $|S_A| \times K$ where K is the number of output units. Thus, the number of parameters is usually larger than that of available data for training in solving linear equalities. To avoid such an underdeterminant situation, pseudo training data $(\hat{\mathbf{c}}_{jl}, \mathbf{z}(\hat{\mathbf{c}}_{jl}))$ for each active RBF are temporally generated by using the network input-output function as follows:

$$\begin{aligned} (\hat{\mathbf{c}}_{jl}, \mathbf{z}(\hat{\mathbf{c}}_{jl})) &= (\mathbf{c}_j + \Delta \mathbf{c}_l, \mathbf{z}(\mathbf{c}_j + \Delta \mathbf{c}_l)) \\ &\text{for } j \in S_A; l = 1, \dots, P_j \end{aligned} \tag{2}$$

where P_j is the number of pseudo data for the j th center \mathbf{c}_j . Here, P_j is determined based on the complexity of the function shape. This complexity can be estimated by the following Hessian information $H(\cdot)$:

$$H(\mathbf{c}_j) = \min \left\{ \frac{\sum_k |\mathbf{h}_k(\mathbf{c}_j)|}{h_0}, 1 \right\} \tag{3}$$

where h_0 is a normalization constant and $|\mathbf{h}_k(\mathbf{c}_j)|$ corresponds to the determinant of the Hessian whose (i, i') -th element is given by

$$\frac{\partial^2 z_k}{\partial c_{ji} \partial c_{j'i'}} = 4 \sum_{j'=1}^J \frac{w_{kj'}}{\sigma_j^2} (c_{ji} - c_{j'i})(c_{j'i'} - c_{j'i}) y_{j'} \tag{4}$$

where w_{kj} is the connection weight from the j th RBF to the k th output and σ_j is the width parameter of the j th RBF. Larger Hessian information means that the approximated function is more complex; thus, the number of pseudo training data P_j is determined as follows:

$$P_j = P_{\max} H(\mathbf{c}_j) \tag{5}$$

where P_{\max} is the maximum number of pseudo training data¹.

2.4 Selection of Modified Connections

Since an RBF has local response to an input domain, the learning is mainly conducted at connections to a limited number of active RBFs. This suggests that the approximation accuracy does not hurt seriously even if the modified connections are restricted to only the connections to active RBFs. To quantify RBF activity, the following contribution index r_j for the j th RBF is defined by the sum of RBF outputs for a training data \mathbf{x}_p and retrieved memory items $\mathbf{M}_{j'}$ ($j' \in S_A$).

$$r_j = \min \left\{ y_j(\mathbf{x}_p) + \sum_{j' \in S_A} y_j(\mathbf{c}_{j'}), 1 \right\} \tag{6}$$

The connections to be modified are limited to the RBFs whose contribution index r_j is larger than a threshold η_3 (i.e., $r_j > \eta_3$). Let us define the index set \tilde{S}_A for these active RBFs by $\tilde{S}_A = \{j \mid r_j \geq \eta_3; j = 1, \dots, J\}$. Then, the modified connections are denoted as $\tilde{\mathbf{W}} = \{\mathbf{w}_j\}_{j \in \tilde{S}_A}$ where \mathbf{w}_j represents the connections between the j th RBF and the outputs.

2.5 Learning of Connections

To update the connection matrix $\tilde{\mathbf{W}}$, the outputs of RBFs in \tilde{S}_A are first calculated for a training data \mathbf{x}_p , the retrieved memory item \mathbf{c}_j ($j \in S_A$), and the pseudo data $\hat{\mathbf{c}}_{jl}$ ($j \in S_A; l = 1, \dots, N_j$). Then, the $(|S_A| + \sum_{j \in S_A} P_j + 1) \times |\tilde{S}_A|$ activation matrix $\tilde{\Phi}$ is defined. Here, $|\tilde{S}_A|$ is the number of RBFs in \tilde{S}_A .

When learning pseudo training data, sometimes the error does not decrease to a satisfied level without modifying the connections to RBFs that are not included in \tilde{S}_A . Empirically, it is known that such pseudo data often exist in

¹ Since a large P_{\max} results in high computation costs, it should be determined depending on available computer resource.

the outermost domain supported by the centers of RBFs in \tilde{S}_A . To alleviate the interference by such pseudo training data, the learning ratio should be small for the connections to the RBFs whose centers are in the outermost domain.

For this purpose, the learning ratio is determined based on the contribution index r_j and define a matrix \mathbf{R} whose diagonal elements are given by r_j ($j \in \tilde{S}_A$). Using \mathbf{R} , the update of connections $\Delta\tilde{\mathbf{W}}$ is given by

$$\Delta\tilde{\mathbf{W}} = \mathbf{R}(\tilde{\Phi}'\tilde{\Phi})^{-1}\tilde{\Phi}'(\mathbf{T} - \tilde{\Phi}\tilde{\mathbf{W}}) \tag{7}$$

where $\tilde{\mathbf{W}}$, $\tilde{\Phi}$, and \mathbf{T} correspond to the matrix of restricted connections, the activation matrix, and the target matrix for $\tilde{\Phi}$, respectively. Instead of calculating the inverse of the matrix in Eq. (7), we use the singular decomposition $\tilde{\Phi} = \mathbf{U}\mathbf{D}\mathbf{V}'$; then, Eq. (7) is reduced to

$$\Delta\tilde{\mathbf{W}} = \mathbf{R}\mathbf{V}\mathbf{D}^{-1}\mathbf{U}'(\mathbf{T} - \tilde{\Phi}\tilde{\mathbf{W}}). \tag{8}$$

3 Experiments

3.1 Study on Negative Interference

Scale of Negative Interference. There are two types of interference caused by learning training data: the one is *positive interference* enhancing the generalization performance and the other is *negative interference* leading to the destruction of past knowledge [10]. To study the effect of the interference, we propose a scale to quantify the positive and negative interference and utilize for evaluating the capability of suppressing the interference in the proposed RAN-LLR.

Let $f(x)$ and $f'(x)$ be the input-output functions of a neural network before and after the incremental learning is carried out, respectively; and let $f^*(x)$ be the true function. Furthermore, let us define the following two sets S_I^+ and S_I^- for a training data set X :

$$S_I^+ = \{x|x \in X, |f(x) - f^*(x)| > |f'(x) - f^*(x)|\} \tag{9}$$

$$S_I^- = \{x|x \in X, |f(x) - f^*(x)| \leq |f'(x) - f^*(x)|\}. \tag{10}$$

S_I^+ and S_I^- respectively represent the sets of points that make the generalization error decreased and increased by learning data x incrementally. Then, the positive interference and the negative interference can be measured by

$$I^+ = \int_{x \in S_I^+} |f'(x) - f(x)|dx \tag{11}$$

$$I^- = \int_{x \in S_I^-} |f'(x) - f(x)|dx. \tag{12}$$

The effectiveness of suppressing the interference is measured by the following ratio between the positive and negative interferences.

$$C = \frac{I^-}{I^+ + I^-}. \tag{13}$$

If C is larger than 0.5, it means that the negative interference dominates over the positive one, resulting in poor performance. On the contrary, if C is less than 0.5, it is considered that stable incremental learning is conducted.

Results. To see how the interference is suppressed in RAN-LLR, the following one-dimensional function approximation problem is applied to RAN-LLR:

$$g(x) = \begin{cases} 4(x - n) & (n \leq x < n + 0.5) \\ -4(x - n - 1) & (n + 0.5 \leq x < n + 1). \end{cases} \quad (14)$$

The domain of x is defined for $n = 0, 1, \dots, 9$ in Eq. (14). 200 training data $(x, g(x))$ are randomly generated and they are given to learn one after another.

The evaluation is made by comparing with the following models: (1) Resource Allocating Network with Global Linear Regression (RAN-GLR) and (2) RAN-LLR without the learning rate \mathbf{R} (RAN-LLR(noR)). In RAN-GLR, no restriction on the connections is imposed in learning. Thus, all the connections are trained with all the memory items when a training data is learned. Since the interference is suppressed almost completely, RAN-GLR gives the target performance for RAN-LLR. On the other hand, RAN-LLR(noR) is adopted to see the effectiveness of introducing \mathbf{R} in Eq. (8); that is, the interference caused by removing \mathbf{R} is investigated.

Figures 1 (a) and (b) demonstrate the time evolutions of the average errors and the negative interference rates C . As seen from Fig. 1, at the early learning stages, C is kept small (i.e., the positive interference dominates) and the average error is quickly dropped. In RAN-GLR, the interference is always suppressed quite effectively and the output error quickly converges. As seen from Fig. 1 (b), the proposed RAN-LLR suppresses the interference fairly well and it has almost the same level of interference as RAN-GLR at the last part of learning stages. Without \mathbf{R} in Eq. (8), the negative interference often dominates in RAN-LLR(noR). Obviously, the reason why the error goes up and down after the 70th learning stage in Fig. 1(a) results from the dominance of the negative interference.

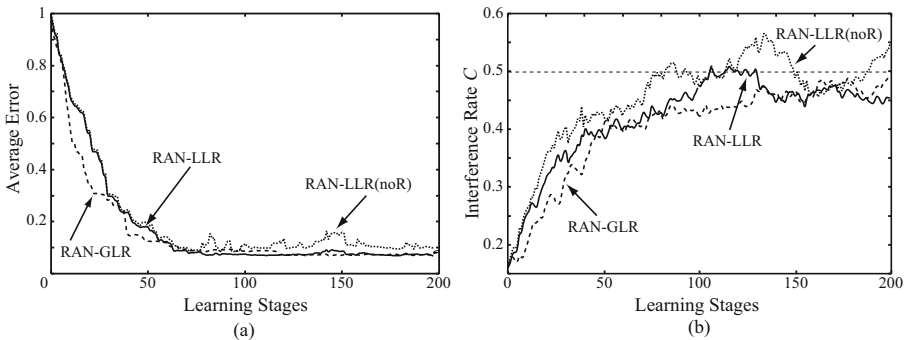


Fig. 1. Time evolutions of (a) average errors and (b) interference rates C

3.2 Performance Evaluation

Experimental Setup. To evaluate the performance of RAN-LLR for noisy and multi-dimensional data, we apply RAN-LLR to the water purification prediction problem where the quantity of injecting chemicals in a filtration plant is predicted. An input is given by a 10-dimensional vector and the output corresponds to the degree of contamination. The numbers of training and test data are 241 and 237, respectively. Training data are randomly given one by one. In RAN and RAN-GD, the learning is terminated when the root mean squared error (RMSE) is lower than 0.1 or the learning times exceeds 50,000.

Here, the following incremental models are evaluated:

1. Locally Weighted Regression (LWR) [9]
2. Memory-based RBF Network (RBFN) [2]
3. Resource Allocating Network (RAN) [4]
4. RAN-LTM by Gradient Descent Learning (RAN-GD) [6]
5. RAN by Global Linear Regression (RAN-GLR) [7]

LWR and RBFN belong to so-called *memory-based learning* models which give high performance but need large memory costs. Therefore, the performance of the two models is referred to the target performance. RAN is adopted to study the effect of suppressing the interference, and RAN-GD is adopted to study the effectiveness of the linear regression approach. To see the effectiveness of restricting modified connections, we also evaluate the performance of RAN-GLR in which all the connections are trained.

Results. Table 1 shows RMSE for test data, learning time, and required memory. As seen from Table 1, the test RMSE of the RBF network is the largest among six algorithms because the learning tends to overfit to training data. In addition, the RBF network requires large memory to store the training data. Although the memory usage of RAN is the smallest and the learning is fast, it has large RMSE due to the lack of the function to suppress the interference. The learning of RAN-GD is also fast but the error is slightly larger than the proposed RAN-LLR. The proposed RAN-LLR has the lowest test RMSE which is almost the same as LWR and RAN-GLR. In terms of learning time and required memory, RAN-LLR has good performance as an incremental learning model.

Table 1. Performances for the water purification prediction problem

	Test RMSE	Time (sec.)	Memory (MB)
LWR	1.66	-	44.3
RBF Net	2.38	75.0	489.7
RAN	2.25	4.5	9.0
RAN-GD	1.93	399.6	13.2
RAN-GLR	1.68	3.9	17.1
RAN-LLR	1.65	2.0	15.0

4 Conclusions

In this paper, we proposed a fast and efficient incremental learning algorithm of RAN-LTM [6] called RAN-LLR in which the linear regression method is applied to learning a restricted set of connections. In the proposed RAN-LLR, only several training data that are essential to maintain the approximation accuracy are selected and stored in long-term memory (LTM) as *memory items*. In order to suppress the catastrophic forgetting, a minimum set of memory items are retrieved from LTM and learned with training data. In addition, pseudo training data are generated based on the complexity of the approximated function and learned to suppress the interference. The primary feature of RAN-LLR is that the computation and memory costs are not significantly increased when a problem domain is dynamically expanded over time.

To evaluate the incremental learning performance of RAN-LLR, we first studied how the negative interference was suppressed to attain good approximation accuracy. Then, we applied RAN-LLR to a real-world prediction problem. As a result, we demonstrated that RAN-LLR had better approximation accuracy compared with the memory-based RBF network and RAN, and that the performance was comparable to LWR with less memory costs.

References

1. Yamauchi, K., Yamaguchi, N., Ishii, N.: Incremental learning methods with retrieving of interfered patterns. *IEEE Trans. on Neural Networks* 10(6), 1351–1365 (1999)
2. Orr, M.: Introduction to radial basis function networks, Tech. Report of Inst. for Adaptive and Neural Computation, Div. of Informatics, Edinburgh University (1996)
3. Haykin, S.: *Neural networks - A comprehensive foundation*. Prentice Hall, Englewood Cliffs (1999)
4. Platt, J.: A resource allocating network for function interpolation. *Neural Computation* 3, 213–225 (1991)
5. Ozawa, S., Toh, S., Abe, S., Pang, S., Kasabov, N.: Incremental learning of feature space and classifier for face recognition. *Neural Networks* 18(5-6), 575–584 (2005)
6. Kobayashi, M., Zamani, A., Ozawa, S., Abe, S.: Reducing Computations in Incremental Learning for Feedforward Neural Network with Long-Term Memory. In: *Proc. of Int. Conf. on Neural Networks 2001*, pp. 1989–1994 (2001)
7. Okamoto, K., Ozawa, S., Abe, S.: A fast incremental learning algorithm of RBF networks with long-term memory. In: *Proc. Int. Conf. on Neural Networks 2003*, pp. 102–107 (2003)
8. Roy, A., Govil, S., Miranda, R.: An algorithm to generate radial basis function (RBF)-like nets for classification problems. *Neural Networks* 8(2), 179–202 (1995)
9. Atkeson, C., Moore, A., Schaal, S.: Locally weighted learning. *Artificial Intelligence Review* 11, 75–113 (1997)
10. Schaal, S., Atkeson, C.G.: Constructive incremental learning from only local information. *Neural Computation* 10(8), 2047–2084 (1998)

Learning Cooperative Behaviours in Multiagent Reinforcement Learning

Somnuk Phon-Amnuaisuk

Perceptions and Simulation of Intelligent Behaviours,
Faculty of Information Technology, Multimedia University,
Jln Multimedia, 63100 Cyberjaya, Selangor Darul Ehsan, Malaysia
somnuk.amnuaisuk@mmu.edu.my

Abstract. We investigated the coordination among agents in a goal finding task in a partially observable environment. In our problem formulation, the task was to locate a goal in a 2D space. However, no information related to the goal was given to the agents unless they had formed a swarm. Further more, the goal must be located by a swarm of agents, not a single agent. In this study, cooperative behaviours among agents were learned using our proposed *context dependent multiagent SARSA* algorithms (CDM-SARSA). In essence, instead of tracking the actions from all the agents in the Q-table i.e., $Q(s, \mathbf{a})$, the CDM-SARSA tracked only actions a_i of agent i and the context c resulting from the actions of all the agents, i.e., $Q_i(s, a_i, c)$. This approach reduced the size of the state space considerably. Tracking all the agents' actions was impractical since the state space increased exponentially with every new agent added into the system. In our opinion, tracking the context abstracted unnecessary details and this approach was a logical solution for multiagent reinforcement learning task. The proposed approach for learning cooperative behaviours was illustrated using a different number of agents and with different grid sizes. The empirical results confirmed that the proposed CDM-SARSA could learn cooperative behaviours successfully.

Keywords: Multiagent reinforcement learning, Context dependent multiagent SARSA, Learning cooperative behaviours.

1 Background

Reinforcement learning is a learning paradigm where feedback to the learner is less specific than feedback in supervised learning; as is the case in many real life scenarios. However, the feedback informs how fruitful the current situation is and the learners could benefit from this feedback. From the literature, many studies of single agent reinforcement learning have been explored, for examples, TD-Gammon, Samuel's checkers player, Acrobat, Elevator dispatching, Dynamic channel allocations and Job-shop scheduling [15]. Recently, multiagent reinforcement learning (MARL) has received much attention. MARL extends a single agent RL to multiagent RL and has attracted the interest of researchers

in game theory [2], [6]; behavioural learning in robots [11]; multi-robot [1], [8]; and multiagent [16] disciplines.

In this paper, our investigation concerns behavioural realisation (i.e., *formation forming*, *formation keeping* and *goal finding*) which is one of the focus areas in robotic swarm and multiagent communities. Swarm formations such as fish schooling, bird flocking and other group hunting behaviours are emerging cooperative behaviours among group members in nature. These behaviours evolve for survival necessity. In nature, group hunting in animals increases the success rate of their hunts. We are interested in emulating this kind of cooperative behaviours in multiagent systems. In our setup, agents must learn to form a group and once the group is formed, the common goal would be realised by all agents. The common goal would disappear if the formation is not kept by all agents. So agents must learn to form a swarm then keep the formation while moving towards the goal. Formation and formation-keeping in robotic swarm have been studied in particle swarm optimisation (PSO) [5] and robotic swarm [1]. Interested readers may want to see a good review on cooperative mobile robotics by [4]. Here, we are interested in investigating these issues from the perspective of multiagent RL.

RL has its roots strongly established around Markov Decision Processes (MDPs). A MDP is a tuple $\langle S, A, R, T \rangle$ where S is a finite discrete set of states, A is a finite set of discrete actions, R is a reward function $R : S \times A \rightarrow \mathfrak{R}$ and T is a state transition function $T : S \times A \rightarrow \Pi(S)$ which informs the probability distribution of all possible next states from the current state-action pair. Among reinforcement learning techniques, Q-learning [17] is one of the most popular techniques for a single agent paradigm. Q-learning learns state-action value to estimate the optimum policy.

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha(R(s, a) + \gamma V(s'))$$

$$V(s) \leftarrow \max_{a \in A} Q(s, a)$$

The extension of Q-learning from a single agent to a multiagent framework has been studied by many colleagues. There are good surveys from [14], [18], [12] and recently by [3] which we would not want to repeat here, though we would touch on some closely related works again for background information.

In the early attempts to apply a standard Q-learning to multiagent by [13], each agent employed different stationary policies. However, the dynamicity of multiagent could not be approximated using stationary policies. It was clear that the state-action pairs in Q-learning must take into account the states and actions of all agents in the system. The other agent's action was taken care of in the minimax-Q-learning suggested by [7]. The minimax-Q-learning was the solution for a zero-sum stochastic game (i.e., competitive behaviours). The minimax-Q-learning was later on extended to Nash-Q that could handle a general sum stochastic game [6]. It was pointed out by [14] that although the minimax-Q and Nash-Q learning algorithms were extended to handle general sum stochastic games, it could be favorable to opt for belief-based algorithms such as *fictitious*

play. In fictitious play, agents maintain their own Q values which were related to joint actions and their past play [18].

In a recent work by [10], a relational representation was proposed to reduce the size of the state space. In his rQ learning approach, an r -state, defined by a set of first order relations, abstracted the states and helped reduce the size of the state space.

All the works mentioned employed variations of Q-learning. In our experiment here, we also resort to Q-learning. We employ SARSA which is an on-policy RL algorithm [15]. In the next section, the proposed *context dependent multiagent SARSA* is elaborated.

2 Context Dependent Multiagent SARSA

Let us quickly go through the main components of a multiagent reinforcement learning: environment, agents, actions and evaluations. The environment (E) in a multiagent setup is always dynamic since an agent's actions are always dependent on other agents' actions. Traditional Markov Decision Process (MDP) for a single agent could be extended to handle many agents by including other agents' actions in the equation (i.e., $Q_i(s, \mathbf{a})$ is the state-action value for agent i on state s and a vector of actions \mathbf{a}):

$$Q_i(s, \mathbf{a}) \leftarrow (1 - \alpha)Q_i(s, \mathbf{a}) + \alpha(R(s, \mathbf{a}) + \gamma V_i(s'))$$

$$V_i(s) \leftarrow \max_{a \in A_i} Q_i(s, a)$$

2.1 Problem Formulation

In our problem formulation, agents must cooperate in order to accomplish the task of goal-finding. The task must be approached in two stages; firstly, agents must form a swarm in order to perceive the common goal and the formation must be kept so that the goal remains visible to them until the task was completed (see figure 1). The environment here was dynamic and the agents' actions were dependent on other agents' actions.

Here, the multiagent SARSA was adopted. Traditionally, $Q_i(s, \mathbf{a})$ is constructed for all the actions of all the agents. This approach results in a huge state space explosion when the number of agents increase. In our implementation, the $Q_i(s, \mathbf{a})$ was set up as $Q_i(s, a_i, c)$ where c was the context. By referring to the context which abstracted agents' actions, the state space was greatly reduced (as compared to keeping track of the agents' actions). In this experiment, there were two contexts, one was *the swarm not yet formed* and the other was *the swarm already formed*. Table 1 summarises our context-dependent multiagent SARSA algorithm.

Determining the context. In our setup, all the agents' actions could be abstracted to whether (i) the joint actions had resulted in the swarm-formation or (ii) not yet. The swarm would be formed if the distances of each agents to the

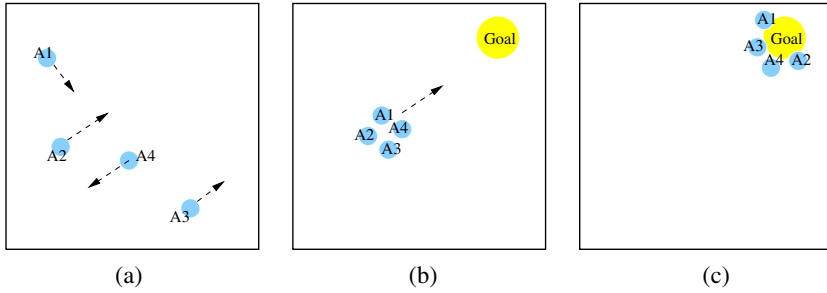


Fig. 1. Problem formulation: (a) agents learned to cooperate and form a swarm, the goal was not informed if a swarm was not formed, (b) the swarm formation was maintained while moving towards a goal, and (c) a goal state

Table 1. The CDM-SARSA algorithm

Context Dependent Multiagent SARSA
Initialise $Q_i(s_i, a_i, c)$ for each agent i arbitrary
Repeat for each episode:
Initialise s_i
Repeat for each step of episode:
Choose a_i according to policy $\pi(s_i)$
Agent i take action a_i
Observe r from s'_i
Observe context c from s'_i
Update value function:
$Q_i(s_i, a_i, c) \leftarrow Q_i(s_i, a_i, c) + \alpha[r + \gamma Q_i(s'_i, a'_i, c') - Q_i(s_i, a_i, c)]$
$s_i \leftarrow s'_i, a_i \leftarrow a'_i, c \leftarrow c'$
Until max step or until the goal is reached
Until max episode

center of the swarm was less than 7 distance units. The center of the swarm was determined by $\sum_i^n Ag_i(x, y)/n$ where n was the number of agents in the model and $Ag_i(x, y)$ denoted the (x, y) coordinates of an agent i .

Determining the reward. Setting up a rewarding mechanism in RL was crucial for the success of RL. The rewarding mechanism determined how the policy was modified. This was directly related to the desired or undesired situation the agent was in. In our experiment, agents were punished if they ran into the wall (-1.0 credit), or moved away from the formation (-0.1 credit); agents were rewarded if they moved towards each other. After agents had successfully formed a swarm, if the center of the swarm moved towards the goal, each agent would be rewarded (0.1 credit). While maintaining the swarm, if agents moved too close to each other (threshold was set at 2 distance unit) they would be punished (-0.1 credit). If the swarm was broken, all agents would be punished (-0.1 credit). All agents were rewarded 100 credits if the goal was achieved.

2.2 Experimental Setup

Environment. The environment here was a 2D landscape with grid sizes of 16×16 , 32×32 and 48×48 (for three sets of experiments). Initialised positions of the agents and a goal were fixed for each experiment (see Table 2). In each time step, an agent might move in any of the following directions: n , e , s , w , ne , nw , se and sw according to its policy. The agents' decisions to move to any one of the eight directions would be evaluated and either a positive or a negative feedback would be given to the agents. The agents, however, were not given any information about the environment and they were supposed to learn the cooperative behaviours by themselves.

Table 2. The parameter settings for the MARL experiments

Parameter Settings	Values
Environment	
Grid size	16×16 32×32 48×48
Position of the Goal	(15,15) (28,28) (44,44)
Position of Agent-1	(2,2) (3,3) (5,5)
Position of Agent-2	(15,2) (30,3) (44,5)
Rewards & Actions	
Formation is formed	0.1 credit
Formation is not formed	-0.1 credit
Move toward goal	0.1 credit
Move away from goal	-0.1 credit
Move closer to formation	0.1 credit
Move away from formation	-0.1 credit
Agents hit the wall	-1.0 credit
The goal is reached	100 credit
Possible actions A	{n, e, s, w, ne, nw, se, sw}
MARL-parameters	
Learning rate α	0.3
Discount rate γ	0.8
ϵ -greedy probability	0.01
Max-iteration	1000
Max-episode	50

Experimental Designs. Two experiments were reported here. In the first experiment, the behaviours of two agents were investigated in three grid sizes. The three grid sizes employed here were 16×16 , 32×32 and 48×48 respectively. Two agents were placed in a specified location (see table 2 for the parameter setting used in the experiments).

In the second experiment, the effects from a different number of agents were investigated. Here, the grid size was fixed at 32×32 and the numbers of agents of 2, 4 and 6 were investigated. The agent Ag_1 to agent Ag_6 were placed at the following coordinates (3,3),(30,3),(6,6),(27,6), (3,30) and (6,27) respectively. The

choice of coordinates for all the experiments was arbitrary but fixed throughout each of the experiment. They were mostly on the corners since the distance would be longest in this way (note, the longer the distance, the harder the problem).

3 Results and Discussion

Locating a goal position in a 2D space by a group of agent is a hard problem when agents are not informed about the goal unless they have formed a swarm. Information about the goal would disappear if the swarm is broken. This induces two kinds of behaviours: forming a swarm and locating the goal which are analogous to group hunting behaviours observed in animals. The constrain of swarm-formation requirement is suitable for studying cooperation in multiagent reinforcement learning. Agents must learn to form a swarm and keep the swarm as this would lead to a more fruitful outcome.

Two sets of experiments were carried out to test our proposed *context-dependent multiagent SARSA*. Behaviours of agents with different grid sizes and with different numbers of agents were illustrated in Figure 2. From the top pane of Figure 2 (experiment I), an average of 280 steps was required for 2 agents to locate the goal for a grid size of 48×48 ; an average of 50 steps for a grid size of 32×32 ; and an average of 20 steps for a grid size of 16×16 . The performance of the system was very impressive since an exhaustive search would require (worst case) of 2304 steps (i.e., 48^2), 1024 steps (i.e., 32^2), and 256 steps (i.e., 16^2) respectively (assuming that two agents had already form a swarm). Without giving away the swarm formation condition, the task might never be accomplished since the probability that all agents would be in the goal position at time t was extremely small.

From the bottom pane of Figure 2 (experiment II), the system's behaviours of different number of agents with a fixed grid size were investigated. It was found that an average of 280 steps had been required for 6 agents to locate the goal for a grid size of 32×32 ; and an average around 50 steps for 4 agents and 2 agents to locate the goal for a grid size of 32×32 . Both experiments confirmed that agents did learn effective policies for the given task (evident from successful policies learned by different agents -see Figure 3; and effectiveness of the learned policies -see Figure 2).

3.1 Discussion on the Approach Taken

SARSA is one of the most popular on-policy temporal difference (TD) methods. SARSA with a single step sample backup is simple to be implemented and yet has proved to be powerful for many single agent applications. In a multiagent scenario, it would not be feasible to apply the traditional SARSA to learn and control agent behaviours since (i) the environments of multiagent are non-stationary, and (ii) the state space would grow exponentially with additional agents added into the system.

Attempts to maintain separate Q-tables for each agent without taking into account other contexts are only logical if all agents do not interact. That is, if

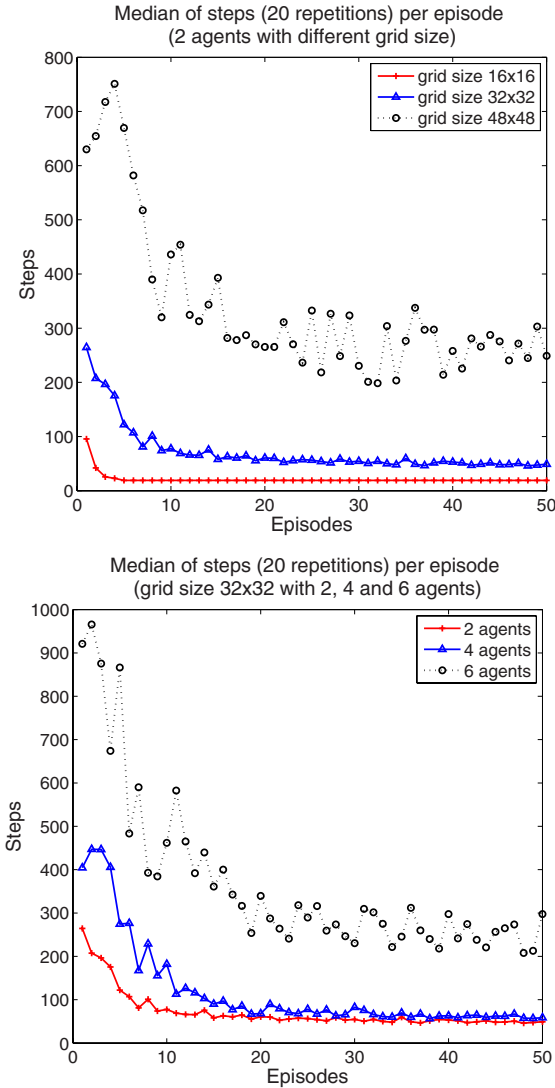


Fig. 2. Experiment I (top): The median of steps (over 20 repetitions) from 50 episodes for two agents. Three different grid sizes employed here were 16×16 , 32×32 and 48×48 ; Experiment II (bottom): The median of steps (over 20 repetitions) from 50 episodes for two, four and six agents. The grid size used in this experiment was fixed at 32×32 .

each agent is independent and coordination/cooperation is not required for the task. Otherwise, it is always necessary to take into account the actions of other agents. What can be done here? Keeping track of all agents actions is a logical extension from the RL to the MARL frameworks. Unfortunately, the state space

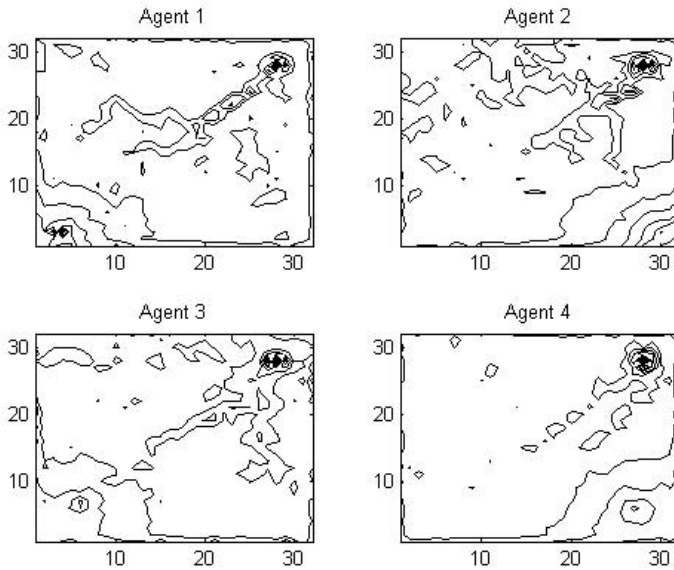


Fig. 3. Gradient of state values from agent 1,2,3,4 (arbitrary picked from one episode). All agents start at different positions and locate the goal at (28,28).

would be intractable soon since each additional agent would expand the state space by the order of $|S|^n|A|^n$ where n is the number of agents, $|S|$ and $|A|$ are the cardinalities of the agents' state-space and actions respectively.

Instead of keeping track of the action vector \mathbf{a} in $Q_i(s, \mathbf{a})$, here we kept track of $Q_i(s_i, a_i, c)$ where c was the context. We argue that with the appropriate context (appropriate here meant the context that reflected the outcome of the dynamicity of the agents' actions), it was possible to abstract a vector of actions \mathbf{a} and replace it with context information c . Figure 2 shows that each agent did learn optimal policies using our proposed CDM-SARSA approach. The goal was located successfully with different grid sizes and with a different number of agents. The gradient of states' value (Figure 3) also confirmed that each agent had learned the path to the goal successfully.

4 Conclusion

It is important to point out that although the cooperative behaviours presented here was only in 2D space, the principle could be generalised and scaled up to real life problems such as unmanned robots in hazardous environments. In such a scenario, it would be inefficient to associate states with spatial positions since there are far too many states. States and actions must be carefully represented as pointed out by [10]. As discussed in this report, it would be a good tactic not to worry about other agents' actions but to focus on the outcome context of the environment from those actions.

We have shown in this paper that adding relevant contexts to reflect the status of the environment (which was altered by other agents' actions) was a good approach in dealing with the explosion of the state space in MARL. In this report, the non-stationary characteristic of the environment was dealt with by adding two contexts to the MARL algorithms instead of adding other agents' actions. These contexts were (i) incomplete formation and (ii) completed formation. The results show that the agents could successfully learn to form a swarm-formation and maintain the formation while searching for the goal. The abstraction of actions to contexts reduced the additional $|\mathcal{S}| \times |\mathcal{S}_i| \times |A_i|$ to only $|\mathcal{S}| \times |C|$, where \mathcal{S} was the current state space and C was the context which $|C|$ was usually a lot smaller than $|\mathcal{S}_i| \times |A_i|$.

In this work, we have shown that it was possible to deal with the dynamicity in MARL by abstracting a vector of actions \mathbf{a} into contexts and then use these contexts in the context dependent multiagent SARSA algorithm. In further work, two directions could be pursued, one is to develop the framework for developing the *context dependent multiagent SARSA* and the other is to apply this idea to deal with intractability issue in a real life problem.

Acknowledgement. I would like to thank anonymous reviewers for their useful comments and suggestions.

References

1. Balch, T., Arkin, R.C.: Behaviour-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation* 14(6), 926–939 (1998)
2. Bowling, M., Velosa, M.: An analysis of stochastic game theory for multiagent reinforcement learning. Technical report, Carnegie Mellon University (2000), <http://www.cs.ualberta.ca/~bowling/papers/00tr.pdf>
3. Bugoniu, L., Babuška, R., Schutter, B.D.: A comprehensive survey of multi-agent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 38(2), 156–172 (2008)
4. Cao, Y.U., Fukunaga, A.S., Kahng, A.B.: Cooperative mobile robotics: Antecedents and directions. *Autonomous Robotics* 4, 1–23 (1997)
5. Duan, H.B., Ma, G.J., Luo, D.L.: Optimal formation reconfiguration control of multiple UCAVs using improved particle swarm optimisation. *Bionic Engineering* 5(4), 340–347 (2009)
6. Hu, J., Wellman, M.P.: Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research* 4, 1039–1069 (2003)
7. Littman, M.L.: Value-function reinforcement learning in markov games. *Journal of Cognitive Systems Research* 2, 67–79 (2001)
8. Matarić, M.J.: Reinforcement learning in multi-robot domain. *Autonomous Robots* 4, 73–83 (1997)
9. Matarić, M.J.: Learning in behaviour-based multi-robot systems: policies, models, and other agents. *Journal of Cognitive Systems Research* 2, 81–93 (2001)
10. Morales, E.F.: Scaling up reinforcement learning with a relational representation. In: *Workshops on Adaptability in Multi-Agent Systems, The First RoboCup Australian Open (AORC 2003)*, Sydney, Australia (January 31, 2003)

11. Morita, M., Ishikawa, M.: Brain-inspired emergence of behaviours based on the desire for existence by reinforcement learning. In: Proceedings of the 15th International Conference on Neural Information Processing (ICONIP 2008), Auckland, New Zealand (2008)
12. Panait, L., Luke, S.: Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems* 11(3), 387–434 (2005)
13. Sen, S., Sekaran, M., Hale, J.: Learning to coordinate without sharing information. In: Proceedings of the 12th National Conference on Artificial Intelligence, pp. 426–431 (1994)
14. Shoham, Y., Powers, R.: Multiagent reinforcement learning: A critical survey. Technical report, Stanford University (2003), http://multiagent.stanford.edu/papers/MALearning_ACriticalSurvey_2003_0516.pdf
15. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. A Bradford Book, The MIT Press (1998)
16. Shoham, Y., Layton-Brown, K.: Multiagent Systems: Algorithmic, Game-theoretic, and Logical Foundations. Cambridge University Press, Cambridge (2009)
17. Watkins, C.J., Dayan, P.: Q-learning. *Machine Learning* 8, 279–292 (1992)
18. Yang, E.F., Gu, D.B.: Multiagent reinforcement learning for multi-robot systems: A survey. Technical report, The University of Essex (2004), <http://cswww.essex.ac.uk/technical-report/2004/cs>

Generating Tonal Counterpoint Using Reinforcement Learning

Somnuk Phon-Amnuaisuk

Music Informatics Research Group,
Faculty of Information Technology, Multimedia University,
Jln Multimedia, 63100 Cyberjaya, Selangor Darul Ehsan, Malaysia
somnuk.amnuaisuk@mmu.edu.my

Abstract. This report discusses the behavioural learning properties of a musical agent learning to generate a two-part counterpoint using *SARSA*, one of the on-policy temporal difference learning approaches. The policy was learned using hand-crafted rules describing the desired characteristics of generated two-part counterpoints. The rules acted as comments about the generated music from a critic. The musical agent would amend its policy based on these comments. In our approach, each episode was a complete 32-bar two-part counterpoint. Form and other contexts (such as chordal context) were incorporated into the system via the critic's rules and the usage of context dependent Q-tables. In this approach the behaviours could be easily varied by amending the critic's rules and the contexts. We provide the details of the proposed approach and sample results, as well as discuss further research.

Keywords: Reinforcement learning, SARSA, Machine generated tonal counterpoint.

1 Background

Reinforcement learning (RL) is a learning paradigm that has been paralleled to the process of reinforcement by dopamine found in humans [1]. The evaluative feedback of an agent's actions in reinforcement learning is not as explicit as the feedback in a supervised learning paradigm. The mapping of states and fruitful actions are usually learned after many trial-and-error attempts. RL has been successfully demonstrated in many tasks such as a pole balancing task where an agent tried to balance a pole as long as it could. In this task, the desired states were states where the pole was balanced in the air and the undesired states were the states where the pole lost its balance. Behavioral learning in this style is suitable to be modelled using reinforcement learning. In this report, we consider the problem of learning creative behaviours, in particular, learning to generate two-part tonal counterpoints.

Algorithmic composition has been investigated by researchers in the AI-music circle since the 1960s. In this area, systems based on rule-based approach, evolutionary approach and machine learning approach have been explored. Among

the variety of approaches that have been investigated by researchers, the applications of RL in the music domain is not widely adopted yet, only recently has reinforcement learning been paid due attention in music domain.

In comparison to rule-based, evolutionary and supervised learning approaches, the RL approach provides flexible more ways to incorporate both exploration and exploitation in its learning mechanism. The exploration of state space in a rule-based approach [5], [9] and an evolutionary approach [7], [15] could be seen as relying on the generate-and-test and modify-and-test techniques respectively. In essence, the rule-based and the evolutionary approaches explore the state space based on a predefined policy residing as rules or fitness functions. This technique could work well in many domains. However, the approaches do not attempt to make use of the experience gained during the search process to adjust the policy. Although, some forms of adaptive search have been investigated, RL offers a different flavour as to how the state space is explored. Past experiences modify an agent's current policies and contribute to the way the state space is explored.

Other machine learning approaches such as artificial neural network [16], [2] and Hidden Markov Model [1] have also been investigated by researchers. These group of machine learning approaches offer attractive features of automated domain knowledge acquisition through training. Nevertheless, the training samples must be prepared, in a way, to manipulate what is supposed to be learned. In this sense, RL offers a more flexible methodology worth investigated into.

Here, we investigated a system that generated compositions in the style of *two-part counterpoint* [4] using RL. In RL paradigm, an agent explores the state space as well as exploits its best knowledge about the terrain of the search landscape. The agent learns about the search landscape from implicit feedback in the RL. Exploration and exploitation in RL are quite unique compared to other approaches.

In music domain, it is generally agreed that it is impossible to lay down fixed rules for composing music. This is because music rules are highly context dependent. However, there are some basic practical norms which novice music students are taught when learning to write a two-part counterpoint. These practical norms could be used to criticise the quality of music generated by an agent. It is possible to adjust an agent's behaviours (i.e., the policy) based on these criticisms. In this way, an optimal policy could be learned by sampling the state space to estimate the utility of state-action pairs $Q(s, a)$:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)]$$

where α is the learning rate and γ is the discount rate (see [13]).

2 Generating Tonal Counterpoints Using SARSA

Counterpoint is the style of music that has more than two or more simultaneous melody lines. Counterpoint has its origin in church music since 900 AD. The term

¹ "...the more common use of the word is that of the combination of simultaneous parts or voices,..." quoted from [8].

counterpoint may carry more specific connotations when used to describe particular stylistics such as *Palestrina counterpoint*, *Bach counterpoint*, etc. Here, the term *tonal counterpoint* is used to describe computer generated music based on *major* and *minor* scales. In this exercise, we are interested to see the agent’s behaviour in generating a two-part *tonal counterpoint* (hereafter, counterpoint) using SARSA [12].

2.1 Knowledge Representation

To apply SARSA to generate a tonal counterpoint, two important criteria must be considered: (i) the states and actions must represent the counterpoint generation process, and (ii) the representation must facilitate the performance evaluation process. A major drawback of RL is the issue of intractable state space. Although this issue is common in all approaches, in RL, it is crucial to devise the representation so that the state space would not be too large. The reason is that RL estimates the optimum policy from all possible states so we would prefer all the states to be visited.

Representing state-action. Here, the states represent the pitches and the actions represent the steps (intervals) between the current pitches and the next pitches. Since we are interested in generating a two-part counterpoint, each state represents a pair of pitches (p_1, p_2) and each action represents a pair of actions (a_1, a_2) , one for each part. Figure 1 illustrates the basic concept of our setup.

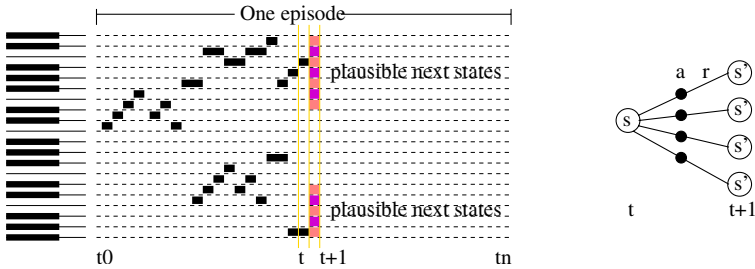


Fig. 1. Formulating two-part counterpoint using SARSA. Each episode, from t_0 to t_n , represents 32 bars of music. At time t the next plausible pitches for the two part are determined using SARSA.

In our two-part counterpoint experiment, the normalised scale degrees in both parts were $\{ \overset{<}{4} \overset{<}{5} \overset{<}{6} \overset{<}{7} \overset{\hat{}}{1} \overset{\hat{}}{2} \overset{\hat{}}{3} \overset{\hat{}}{4} \overset{\hat{}}{5} \overset{\hat{}}{6} \overset{\hat{}}{7} \overset{>}{1} \}$. Each voice could stay the same or move up/down up to five scale degrees $\{-5 -4 -3 -2 -1 0 +1 +2 +3 +4 +5\}$. The representation choice of states and actions above was quite effective for our problem setup. The size of the Q-table would be $12 \cdot 12 \cdot 11 \cdot 11 = 17424$ states.

² $\overset{<}{5}$ means dominant degree in the lower octave and $\overset{>}{5}$ means dominant degree in the upper octave respectively.

Representing performance evaluation criteria. The performance of RL depends on the performance evaluation criteria. Important dimensions in music could be described using *form*, *melody*, *harmony* and *texture*. To generate a two-part counterpoint from $Q^\pi(s, a)$ above, two issues need to be addressed: (i) the evaluation criteria for the generated content, and (ii) the overall form of the composition. Table 1 summarises all the evaluation criteria used in this experiment (see [14] for definitions of these musical terms).

Table 1. Evaluation criteria

Criteria	Reward value
Parallel fifth, octave	-0.1
Crossing between parts	-0.1
Spacing between voice more than one octave	-0.1
Repeated notes	-0.1
Repeated consonant major, minor third	-0.1
Repeated consonant major, minor sixth	-0.1
Wide leap interval	-0.1
Dissonant progression second, tritone	-0.1
Consonant progression major, minor third	0.1
Contrary motion	0.1

For the second issue, we decided to code the desired formal structure of a two-part counterpoint in each episode. That one episode represented a 32 bar composition of a two-part counterpoint. Figure 2 illustrates the tactic we employed to capture both the form and context of the composition. Adding chord (tonality or chordal) contexts could be dealt with by expanding the Q-tables from one to seven tables. The total states would be $17424 \cdot 7 = 121968$ states.



Fig. 2. Coding a desired form and choral contexts into a two-part counterpoint. The figure shows 32 bars chordal context in a major mode.

2.2 Applying SARSA with Chordal Context

SARSA has its name from its backup diagram (i.e., $s, a \xrightarrow{r} s', a'$). It is a popular on-policy TD control. An agent learns the action-value function $Q(s, a)$ from

action a on the state s . Q-learning learns state-action value and estimates the optimum policy using the equations below:

$$Q_c(s, a) \leftarrow Q_c(s, a) + \alpha[r + \gamma Q_c(s', a') - Q_c(s, a)]$$

where α is the learning rate, γ is the discount rate, and c refers to *the chordal context* (see Table 3 for parameter settings in this experiment).

Table 2. The application of SARSA algorithm in this experiment

SARSA	
Initialise $Q_c(s, a)$ for all possible contexts C arbitrary	
Repeat for each episode:	
Initialise s for each Q_c	
Repeat for each step of episode:	
Choose a according to policy $\pi(s)$ and context c	
Agent takes action a	
Observe r from s', a', c'	
Update value function:	
$Q_c(s, a) \leftarrow Q_c(s, a) + \alpha[r + \gamma Q_c(s', a') - Q_c(s, a)]$	
$s \leftarrow s', a \leftarrow a', c \leftarrow c'$	
Until max step or until termination	
Until max episode	

Table 3. The parameter settings in our experiment

SARSA Parameter Settings		
	Figure 4 (a)	Figure 4 (b)
Learning rate (α)	0.3, 0.5, 0.7	0.5
Discount rate (γ)	0.9	0.9
(ϵ)-greedy probability	0.1	0.05, 0.1, 0.2
Max-iteration	512	512
Max-episode	120	120

3 Results and Discussion

It is always hard to find an objective evaluation criteria for computer generated music (or other algorithmic art such as paintings). Here, we present the output from the system which has many interesting characteristics. Figure 3 shows the two-part counterpoint in a G-major mode This is just an example from many plausible outcomes. The generated counterpoint has 32-bar in $\frac{4}{4}$ time which is equivalent to 512 steps in each episode. Due to limited space, only one example of the two-part counterpoint in a major mode is presented here.

In our opinion, the piece is quite tuneful with a good exploitation of an ascending scale motive. There are too many repeated notes here and there but on the

Tonal Counterpoint #1



Fig. 3. The generated counterpoint was in $\frac{4}{4}$ time where 32-bar of music was equivalent to 512 steps in each episode (i.e., each step represented a semi-quaver duration, each bar in $\frac{4}{4}$ time had 16 steps, therefore 32 bars had 512 steps)

whole, the piece does express characteristics of counterpoint compositions. To be more objective about the output, we argue that the system did learn from its experience. The overall signature of this composition emerged from the rewards scheme provided (see Table [II](#)). The rewards and punishments were set according

to our theoretical knowledge of tonal counterpoints. Desired behaviours would be encouraged by positive feedback and undesired behaviours would be discouraged by negative feedback. The magnitude of feedback was set arbitrarily, here, -0.1 to undesired behaviours and 0.1 to desired behaviours. These reward signals r , the learning rate α , and the discount rate γ were used to update agents' policies. Hence, desired behaviours would be encouraged while undesired behaviours would be discouraged in the future.

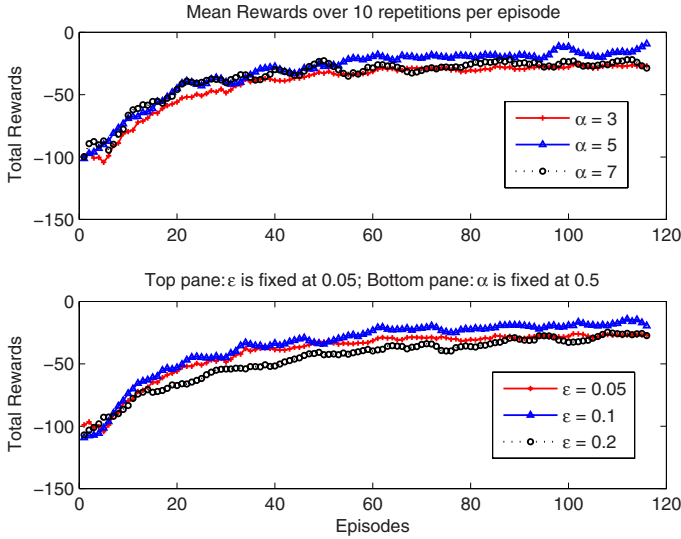


Fig. 4. Smoothed reward values from 120 episodes, (top) varying learning rates α , and (bottom) varying ϵ -greedy probability

From our experiment, the SARSA always converged (see Figure 4). This implies that the model did learn the optimal policy. Figure 5 shows the trend of different reward criteria over 150 episodes. It is evident that the system learned from their past mistakes (by abstaining from repeating the mistakes) as well as promoting fruitful actions (increasing desired behaviours).

3.1 Relations to Previous Work

Unfortunately, we could not really compare the output of this work with other related works in the field. This is mainly because there are so many distinctive features in each work and a comparison to their unique results would not bring any conclusive viewpoint. Instead, a summary of closely related works is given and discussed below.

In [6], a hybrid of recurrent artificial neural network and the actor-critic temporal difference method were explored in a jazz improvisation task. In [3],

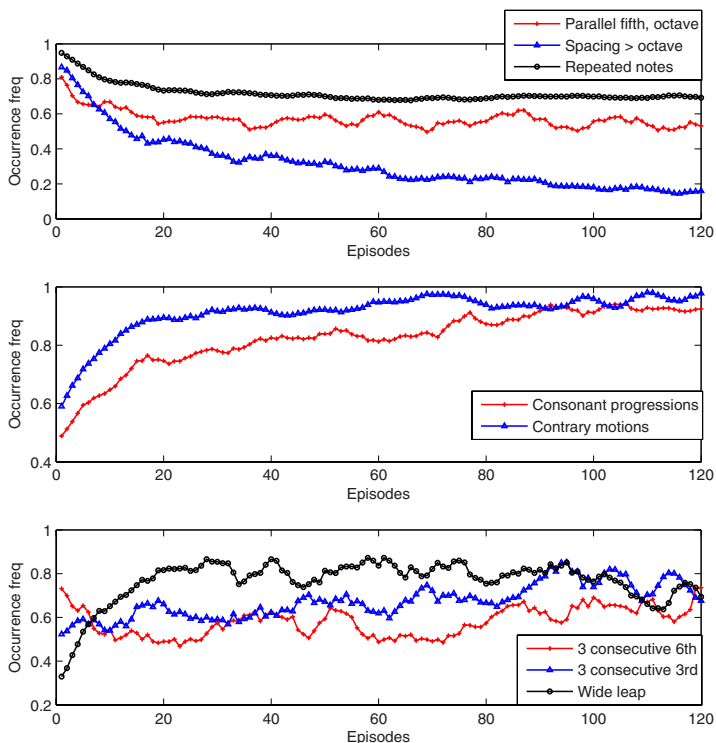


Fig. 5. Changes in behaviours over 120 episodes; (top pane) reduce in undesired behaviours; (middle pane) increase in desired behaviours; (bottom pane) fluctuation in consecutive third, sixth, and wide leap progressions

Dyna-Q RL was employed in the study of *automatic improvisation* and *style imitation*. In a recent report by [4], SARSA(λ) was employed to create an *automated interactive music*.

In RL paradigm, the representation of states, actions and reinforcement signals are the important components. The application of RL in the music domain in previous works, commonly, abstract music to pitch, and duration of pitch as basic building blocks. Other information, e.g., melodic interval, harmonic intervals, melodic contours, harmonic movements, rhythmic motives, etc., may be further derived from those basic building blocks.

In our work, similar abstraction was employed for music knowledge representation. The SARSA was also employed in our experiment. However, to reduce the size of the state space, states and actions were represented using scale degrees and the number of steps up/down the scale degrees. Other derived features such as harmonic intervals and melodic contours were organised in terms of rules and contexts (e.g., chordal contexts).

4 Conclusion and Further Work

Policy learning in RL is a powerful concept. An agent is left to explore a partially observable environment until it learns a policy (i.e., how it should react to the environment) that maximises its return, R . The representation of the state space, \mathcal{S} , and actions, \mathcal{A} , are critical since they are the abstraction of behaviours to be learned.

Temporal-difference learning is an effective technique for learning a policy. In this work, we explored SARSA which was a variant of TD learning to generate 32-bar two-part counterpoint pieces. By carefully selecting the representation of states, actions, rules and contexts, a complex problem such as algorithmic composition could be dealt with and reasonable output is obtained with comparatively less effort. Our approach could potentially facilitate mass automated music generation where each composition could still be uniquely conditioned by a set of rules, form and context provided to each piece. In further work, a few immediate directions could be pursued from here: (i) to improve the handcrafted rules for different composition, (ii) to automate rules-acquisition process, and (iii) to apply the approach to other genres (e.g., four part writing, jazz, etc).

Acknowledgement. The author would like to thank anonymous reviewers for their useful comments.

References

1. Allan, M., Williams, C.K.: Harmonising chorales by probabilistic inference. In: Saul, L., Weiss, Y., Bottou, L. (eds.) *Advances in Neural Information Processing Systems*, vol. 17. MIT Press, Cambridge
2. Chen, C.C.J., Miikkulainen, R.: Creating melodies with evolving recurrent neural networks. In: *Proceedings of the 2001 International Joint Conference on Neural Network, IJCNN 2001*, Washington DC. IEEE, Los Alamitos (2001)
3. Cont, A., Dubnov, S., Assayag, G.: Anticipatory model of musical style imitation using collaborative and competitive reinforcement learning. In: Butz, M.V., Sigaud, O., Pezzulo, G., Baldassarre, G. (eds.) *ABiALS 2006*. LNCS (LNAI), vol. 4520, pp. 285–306. Springer, Heidelberg (2007)
4. Collins, N.: Reinforcement learning for live musical agents. In: *Proceedings of the International Computer Music Conference, ICMC 2008*, Belfast, Ireland, August 24–29 (2008)
5. Ebcioglu, K.: An expert system for harmonizing four-part chorales. In: Balaban, M., Ebcioglu, K., Laske, O. (eds.) *Understanding Music with AI: Perspectives on music cognition*, Ch.12, pp. 294–333. The AAAI Press/The MIT Press
6. Franklin, J.A., Manfredi, V.U.: Nonlinear credit assignment for musical sequences. In: *Second International Workshop on Intelligent System Design and Application*, pp. 245–250 (2002)
7. Horner, A., Goldberg, D.E.: Genetic algorithms and computer-assisted music composition. In: Belew, R., Booker, L. (eds.) *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kauffman, San Francisco (1991)
8. Kennedy, M.: *The Concise Oxford Dictionary of Music*. Oxford University Press, Oxford (1996)

9. Phon-Amnuaisuk, S.: Control language for harmonisation process. In: Anagnostopoulou, C., Ferrand, M., Smaill, A. (eds.) *ICMAI 2002*. LNCS (LNAI), vol. 2445, p. 155. Springer, Heidelberg (2002)
10. Saul, L.K., Jordan, M.I.: Mixed memory markov models: Decomposing complex stochastic processes as mixtures of simpler ones. *Machine Learning* 37(1), 75–87 (1999)
11. Schultz, W.: Predictive reward signal of dopamine neurons. *Journal of Neurophysiology* 80, 1–27 (1998)
12. Sutton, R.S.: Generalization in reinforcement learning: Successful examples using sparse course coding. In: Touretzky, D.S., Mozer, M.C., Hasselmo, M.E. (eds.) *Proceedings of the Advances in Neural Information Processing Systems Proceedings*, pp. 1038–1044. MIT Press, Cambridge (1996)
13. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*. The MIT Press, A Bradford Book (1998)
14. Taylor, E.: *The AB Guide to Music Theory (part I and part II)*. The Associated Board of the Royal Schools of Music (1989)
15. Todd, P.M., Werner, G.M.: Frankensteinian methods for evolutionary music composition. In: Griffith, N., Todd, P.M. (eds.) *Musical Networks: Parallel Distributed Perception and Performance*, pp. 313–340. The MIT Press, Cambridge
16. Toiviainen, P., Eerola, T.: A method for comparative analysis of folk music based on musical feature extraction and neural networks. In: *VII International Symposium on Systematic and Comparative Musicology and III International Conference on Cognitive Musicology*, University of Jyväskylä, Finland, August 16-19 (2001)
17. Watkins, C.J., Dayan, P.: Q-learning Machine. *Learning* 8, 279–292 (1992)

Robust Approximation in Decomposed Reinforcement Learning

Takeshi Mori and Shin Ishii

Kyoto University, Gokasho, Uji, Kyoto, 611-0011, Japan
{tak-mori, ishii}@i.kyoto-u.ac.jp

Abstract. Recently, an efficient reinforcement learning method has been proposed, in which the problem of approximating the value function is naturally decomposed into a number of sub-problems, each of which can be solved at small computational cost. While this method certainly reduces the magnitude of temporal difference error, the value function may be overfitted to sampled data. To overcome this difficulty, we introduce a robust approximation to this context. Computer experiments show that the value function learning by our method is much more robust than those by the previous methods.

Keywords: reinforcement learning, approximation of value function, robust approximation.

1 Introduction

In many realistic reinforcement learning (RL) problems, large state and action spaces make the value function estimation in its original function space impractical. One possible idea to deal with this problem is to decompose the value function effectively. In [5], the conventional RL problem of estimating the value function was reformulated into that of estimating the error in the value function. The error, i.e., the decomposed value function, can be more easily approximated than the value function itself. Hence, an efficient RL method that approximates the errors by a least-squares (LS) method and reconstructs the value function by combining those errors was proposed [5].

Although this method was successful in minimizing the magnitude of temporal difference (TD) errors effectively, the value function can be overfitted into sampled data, because the LS optimization is likely to be disturbed by outliers. To overcome this difficulty, in this study, we introduce a robust approximation to the decomposed RL [5]. That is, we employ the least-absolute deviation (LAD) technique [3] instead of the LS optimization. Computer experiments show that the value function learning by our method is more robust than those by the previous methods [5] [4].

2 MDPs and Value Function Approximation

We consider finite Markov decision processes (MDPs). At time t , the agent selects an action a_t according to a stationary policy π at a state s_t , and then moves to

a next state s_{t+1} and simultaneously receives a reward r_{t+1} . The objective is to find the policy that maximizes the action value function:

$$Q^\pi(s, a) = E_\pi \left[\sum_{i=t}^{\infty} \gamma^{i-t} r_{i+1} \mid s_t = s, a_t = a \right], \tag{1}$$

where $\gamma \in [0, 1)$ is a discount factor.

One approach to seeking an optimal policy is policy iteration [6]. This is composed of two steps, i.e., a policy evaluation step and a policy improvement step. In the former, the value function Q^π for the current policy π is calculated or approximated. In the latter, the policy π is improved based on the learned value function $\hat{Q}^\pi (\approx Q^\pi)$. In this study, we focus on the former problem. The Bellman equation under π is defined as

$$Q^\pi(s, a) = E_\pi[r_{t+1} + \gamma Q^\pi(s_{t+1}, a_{t+1}) \mid s_t = s, a_t = a]. \tag{2}$$

In realistic RL problems, the value function is often approximated by using a parametric linear model, i.e., it is represented as a linear combination of basis functions $\phi(s, a)$ whose linear coefficients constitute the parameter θ :

$$Q^\pi(s, a) \approx \sum_{m=1}^M \phi_m(s, a) \theta_m \equiv \phi(s, a)' \theta, \tag{3}$$

where ($'$) is the transpose. Note that the designer of the learning system must prepare the basis functions prior to learning. The parameter θ is adjusted in the policy evaluation step by using the sample trajectory. One of the learning methods is least-squares TD learning (LSTD) [2], which obtains a closed-form solution for the linearly-approximated value function. In LSTD, for an observed sample trajectory, the parameter vector θ is optimized so as to minimize the cost function:

$$J_{LSTD}(\theta) = \frac{1}{2} \sum_{t=0}^{T-1} (Q^\pi(s_t, a_t) - \phi(s_t, a_t)' \theta)^2, \tag{4}$$

where T is the trajectory length. The vector θ is obtained by the least-squares optimization; more concretely [2],

$$\theta = \mathbf{A}^{-1} \mathbf{b}, \tag{5}$$

where matrix \mathbf{A} and vector \mathbf{b} are given by

$$\mathbf{A} = \sum_{t=0}^{T-1} \phi(s_t, a_t) (\phi(s_t, a_t) - \phi(s_{t+1}, a_{t+1}))' \tag{6}$$

$$\mathbf{b} = \sum_{t=0}^{T-1} \phi(s_t, a_t) r_{t+1}, \tag{7}$$

respectively.

3 Decomposed Reinforcement Learning

According to the idea of decomposed RL [5], on the contrary, the value function is repeatedly updated as

$$\hat{Q}^\pi(s, a) := \hat{Q}^\pi(s, a) + \psi(s, a)' \boldsymbol{\theta}. \tag{8}$$

In each update, the basis function $\psi(s, a)$ and the parameter $\boldsymbol{\theta}$ are updated in order to approximate the error $Q^\pi(s, a) - \hat{Q}^\pi(s, a)$,

$$Q^\pi(s, a) - \hat{Q}^\pi(s, a) \approx \psi(s, a)' \boldsymbol{\theta}, \tag{9}$$

instead of the value function itself (Eq.(3)). The basis function $\psi(s, a)$ is determined by the minimax approximation [1], and the parameter $\boldsymbol{\theta}$ is determined by the procedure shown below.

The cost function of the least-squares optimization of $\boldsymbol{\theta}$ is given by

$$J_{dLSTD}(\boldsymbol{\theta}) = \frac{1}{2} \sum_{t=0}^{T-1} (Q^\pi(s_t, a_t) - \hat{Q}^\pi(s_t, a_t) - \psi(s_t, a_t)' \boldsymbol{\theta})^2. \tag{10}$$

The derivative of the cost function with respect to $\boldsymbol{\theta}$ becomes

$$\nabla_{\boldsymbol{\theta}} J_{dLSTD}(\boldsymbol{\theta}) = \sum_{t=0}^{T-1} \psi(s_t, a_t) (\hat{U}^\pi(s_t, a_t) - \psi(s_t, a_t)' \boldsymbol{\theta}). \tag{11}$$

The target of the regression problem above, $\hat{U}^\pi(s_t, a_t) \equiv Q^\pi(s_t, a_t) - \hat{Q}^\pi(s_t, a_t)$, is replaced by the expectation of the discounted cumulative TD error:

$$\hat{U}^\pi(s, a) = E_\pi \left[\sum_{i=0}^{\infty} \gamma^i \delta_{t+i+1} \mid s_t = s, a_t = a \right], \tag{12}$$

where δ_{t+1} is the TD error:

$$\delta_{t+1} \equiv r_{t+1} + \gamma \hat{Q}^\pi(s_{t+1}, a_{t+1}) - \hat{Q}^\pi(s_t, a_t). \tag{13}$$

From its definition, Eq.(12), $\hat{U}^\pi(s, a)$ satisfies

$$\hat{U}^\pi(s, a) = E_\pi [\delta_{t+1} + \gamma \hat{U}^\pi(s_{t+1}, a_{t+1}) \mid s_t = s, a_t = a], \tag{14}$$

which corresponds to the Bellman equation for $\hat{U}^\pi(s, a)$. Because $\hat{U}^\pi(s, a)$ is unknown, a bootstrapping technique [6] is employed to approximate $\hat{U}^\pi(s, a)$ in Eq.(11) such that the target $\hat{U}^\pi(s_t, a_t)$ is replaced by $\delta_{t+1} + \gamma \psi(s_{t+1}, a_{t+1})' \boldsymbol{\theta}$, and then the gradient (11) is replaced by

$$\tilde{\nabla}_{\boldsymbol{\theta}} J_{dLSTD}(\boldsymbol{\theta}) \equiv \sum_{t=0}^{T-1} \psi(s_t, a_t) (\delta_{t+1} - (\psi(s_t, a_t) - \gamma \psi(s_{t+1}, a_{t+1}))' \boldsymbol{\theta}). \tag{15}$$

The solution is given from $\tilde{\nabla}_{\theta} J_{dLSTD}(\theta) = \mathbf{0}$ as a closed form:

$$\theta = \mathbf{B}^{-1} \mathbf{c}, \tag{16}$$

where

$$\mathbf{B} = \sum_{t=0}^{T-1} \psi(s_t, a_t) (\psi(s_t, a_t) - \gamma \psi(s_{t+1}, a_{t+1}))' \tag{17}$$

$$\mathbf{c} = \sum_{t=0}^{T-1} \psi(s_t, a_t) \delta_{t+1}. \tag{18}$$

This policy evaluation method is called “differential LSTD” (dLSTD) [5].

4 Robust Approximation in Decomposed RL

In dLSTD, the value function is repeatedly updated by adding a parameterized model (Eq. (8)) to represent well the residual of the value function based on the sampled data; this iterative procedure may make the value function overfitted to the sampled data, in comparison to optimization of a single regression model such as LSTD (Eqs. (5)-(7)). Especially when the number of samples is small or their variance is large, the quality of the dLSTD value function gets worse as the number of updating the value function (Eq. (8)) increases. This is a general weak point of LS methods. For example, suppose that we are seeking the mean parameter from five samples: $\{x_0 = 1.8, x_1 = 1.9, x_2 = 2.0, x_3 = 2.3, x_4 = 100\}$, where x_0, \dots, x_3 are from the true distribution of mean of 2.0, but x_4 is an outlier. Based on the LS cost function:

$$H_{LS}(\theta) = \frac{1}{2} \sum_{m=0}^4 (x_m - \theta)^2, \tag{19}$$

the parameter θ is estimated as 21.6 as the solution of $\partial H_{LS}(\theta) / \partial \theta = 0$. Due to the outlier x_4 , this estimation is much different from the true parameter value 2.0.

On the other hand, the cost function of the LAD optimization [3] of θ is given based on the L_1 norm:

$$H_{LAD}(\theta) = \sum_{m=0}^4 |x_m - \theta|, \tag{20}$$

where $|\cdot|$ denotes the absolute value. The parameter estimate is given by setting the derivative at 0:

$$\begin{aligned} \frac{\partial}{\partial \theta} H_{LAD}(\theta) &= \sum_{t=0}^4 \text{sign}(x_t - \theta) = 0 \\ \Rightarrow \hat{\theta} &= \text{median}\{x_0, \dots, x_4\} = 2.0, \end{aligned} \tag{21}$$

where $\text{sign}(\cdot)$ returns the sign of the input. The obtained estimate $\hat{\theta}$ is 2.0, which is much robust to the outlier x_4 .

By employing this technique, the cost function of dLSTD (Eq. (10)) is modified to

$$J_{dLADTD}(\theta) = \sum_{t=0}^{T-1} |Q^\pi(s_t, a_t) - \hat{Q}^\pi(s_t, a_t) - \psi(s_t, a_t)' \theta|. \tag{22}$$

The derivative of the cost function (22) with respect to θ becomes

$$\nabla_{\theta} J_{dLADTD}(\theta) = \sum_{t=0}^{T-1} \psi(s_t, a_t) \text{sign}(\hat{U}^\pi(s_t, a_t) - \psi(s_t, a_t)' \theta). \tag{23}$$

Because $\hat{U}^\pi(s, a)$ is unknown, we introduce a bootstrapping technique [6]. Similar to Eqs. (12)-(15), we obtain the approximate derivative:

$$\tilde{\nabla}_{\theta} J_{dLADTD}(\theta) \equiv \sum_{t=0}^{T-1} \psi(s_t, a_t) \text{sign}(\delta_{t+1} + \gamma \psi(s_{t+1}, a_{t+1})' \theta - \psi(s_t, a_t)' \theta).$$

Because the sign function is nonlinear and $\tilde{\nabla}_{\theta} J_{dLADTD}(\theta) = \mathbf{0}$ cannot be solved in a closed form, we perform gradient-based optimization of the objective function. Then, the parameter θ is estimated by repeating

$$\theta := \theta - \alpha \tilde{\nabla}_{\theta} J_{dLADTD}(\theta), \tag{24}$$

where α is a positive learning coefficient. We call this method “differential least-absolute deviation temporal difference learning” (dLADTD).

Pseudo-code of dLADTD in policy iteration

(Sampling phase) Generate a sample trajectory

(Learning phase) Repeat

1. Calculate the TD error: $\delta_0, \dots, \delta_{T-1}$
2. Generate the new basis function $\psi(s, a)$ based on $\delta_0, \dots, \delta_{T-1}$ by [1].
3. Repeat
 - Calculate the approximate gradient $\tilde{\nabla}_{\theta} J_{dLADTD}(\theta)$
 - Update the parameter θ as $\theta := \theta - \alpha \tilde{\nabla}_{\theta} J_{dLADTD}(\theta)$
4. Update the value function: $\hat{Q}^\pi(s, a) := \hat{Q}^\pi(s, a) + \psi(s, a)' \theta$

(Policy update phase) Update policy and go back to (Sampling phase) or (Learning phase)

5 Computer Experiments

We compared our method with the decomposed RL [5] and other comparable LSTD-based method [4] by using a 200-state chain problem.

5.1 Experimental Settings

In this problem, an agent moves through a single chain consisting of 200 states, and the agent’s objective is to maximize the expected return defined as accumulated discount rewards. In each time step, the agent selects an action from two candidates, “left” and “right”. Each action is successfully done with probability 0.9, leading to state transition in the intended direction, but fails with probability 0.1, making the state change in the opposite direction. The two boundaries of the chain are dead-ends where the outward transition is replaced by staying at the same state. Eight states are selected randomly from all of the 200 states in each learning run; a reward of 1 is given when the agent is at one of the selected states, otherwise, no reward (0) is given. The discount factor is set at $\gamma = 0.9$.

First, as a sampling phase, we generated a single trajectory whose length was $T = 5,000$ by using a random policy. Next, we performed a learning phase through which we evaluated our method. In each step in the learning phase, 2-basis functions was produced by the minimax method [11], such that $\psi(s, a) = [1, 0]'$ when the TD error of the pair (s, a) was larger than its average or $\psi(s, a) = [0, 1]'$ otherwise. The policy was updated when log-RMSE (logarithmic root mean squares error) of Q -function (see the next subsection) became -0.8 . The entire RL algorithm was shown in the previous section.

5.2 Experimental Results

We compared our method (dLADTD), decomposed RL (dLSTD) [5], comparable LSTD-based method (LSTD) [4], and the LSTD-based method with the random basis function (LSTD-*random*) in which $\psi(s, a) = [1, 0]'$ or $[0, 1]'$ was randomly chosen regardless of the values of the state and action. We implemented them using matlab 7.7.0(R2008b) on 3.00 GHz Intel(R) Xeon(R). Fig 1 and Fig 2 show the learning curves averaged over 10 learning runs for the 200-chain problem, where Fig 1 shows the comparison in the policy evaluation and Fig 2 in the policy iteration. The vertical axes in Fig 1(a) denote log RMSTDE (logarithmic root mean square TD error) averaged over all pairs of state and action: $\|\delta\| \equiv \log \sqrt{\sum_{t=0}^{T-1} \delta_{t+1}^2 / N}$, where N is the number of samples. The vertical axes in Fig 1(b) denote log RMSE (the logarithmic root mean square error) between Q^π and \hat{Q}^π : $\|Q^\pi - \hat{Q}\| \equiv \log \sqrt{\sum_{t=0}^{T-1} (Q^\pi(s_t, a_t) - \hat{Q}^\pi(s_t, a_t))^2 / N}$. The vertical axes in Fig 2 denote log RMSE between $Q^{\pi^*}(s, a)$ and $Q^\pi(s, a)$, where Q^{π^*} is the value function under the optimal policy π^* : $\|Q^{\pi^*} - Q^\pi\| \equiv \log \sqrt{\sum_{t=0}^{T-1} (Q^{\pi^*}(s_t, a_t) - Q^\pi(s_t, a_t))^2 / N}$. The horizontal axes in the upper panels of Figs 1 and 2 denote the iteration steps, and those in the lower panels denote the CPU time (sec.).

In Fig 1(a), we can see that dLSTD is much faster than the other methods in terms of the CPU time. This result supports the conclusion of [5] about the effectiveness of the value function decomposition. However, the TD error

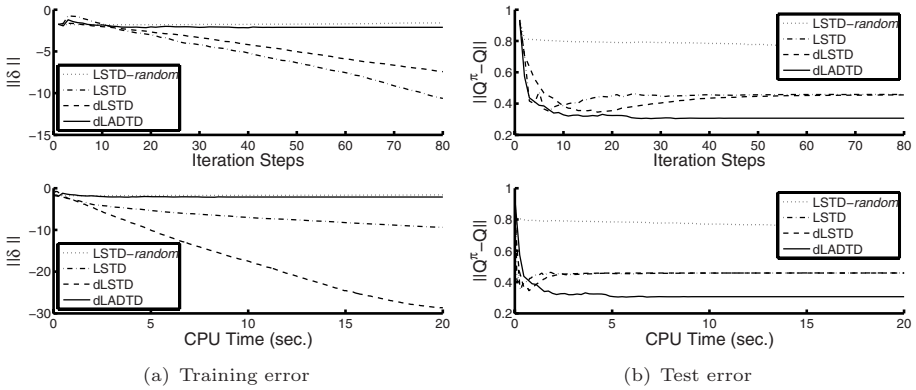


Fig. 1. Policy evaluation

can be seen as the training error, which represents the approximation of the value function for the sampled data, not for new data. Practical learning algorithms should pay attention to the test error rather than the training error.

Fig. 1(b) shows the test error. Although the learning curve of our new algorithm (dLADTD) was much worse than those of the dLSTD and LSTD in Fig. 1(a), it showed the best generalization ability in Fig. 1(b). This is due to the fact that the negative effects of outliers and large sampling variance have actually eased by the LAD optimization. The learning speed of dLADTD was slightly slower than the dLSTD and LSTD in terms of CPU time, because the iterative gradient-based procedure (Eq. (24)) are in addition necessary in dLADTD.

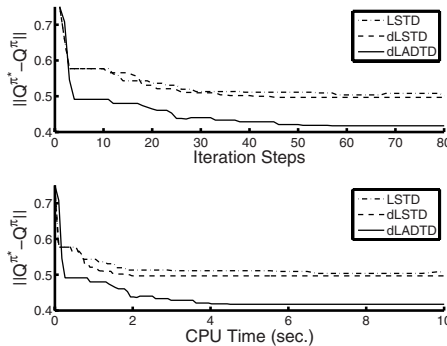


Fig. 2. Policy iteration

Similar to Fig. 1, the best performance was achieved by the dLADTD in policy iteration (Fig. 2). The policy acquired by the dLADTD was much nearer to the optimal policy than those by the other LSTD-based methods.

6 Conclusion

In this study, we proposed a novel scheme for policy evaluation, where the value function was estimated by sequentially approximating its approximation error, by means of the robust regression (LAD) technique. The value function was more robustly approximated by LAD than by the previous least squares-based methods [5] [4]. Our experiments showed that our method has better performance in the generalization of the value function than the existing LSTD-based methods.

References

1. Bertsekas, D., Castañón, D.: Adaptive aggregation methods for infinite horizon dynamic programming. *IEEE Transactions on Automatic Control* 34, 589–598 (1989)
2. Bradtke, S.J., Barto, A.G.: Linear least-squares algorithms for temporal difference learning. *Machine Learning* 22(2), 33–57 (1996)
3. Dasgupta, M., Mishra, S., Hill, N.: Least absolute deviation estimation of linear economic models: A literature review. *Munich Personal PePEc Archive* 1781 (2004)
4. Keller, P.W., Mannor, S., Precup, D.: Automatic basis function construction for approximate dynamic programming and reinforcement learning. In: *Proceedings of the Twenty-third International Conference on Machine Learning* (2006)
5. Mori, T., Ishii, S.: An additive reinforcement learning. In: *The Nineteenth International Conference on Artificial Neural Networks* (2009)
6. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*. MIT Press, Cambridge (1998)

Learning of Go Board State Evaluation Function by Artificial Neural Network

Hiroki Tomizawa, Shin-ichi Maeda, and Shin Ishii

Graduate School of Informatics, Kyoto University, Japan
{tomizawa-h, ichi, ishii}@sys.i.kyoto-u.ac.jp

Abstract. We construct an artificial neural network called T361G to evaluate Go board state (expected winning probability of Black's/White's win conditioned on the current board state in Black's/White's turn). Different from the existing Monte-Carlo Go [3][4], which evaluates the next move (the next board state) by performing random simulations in every turn, we use a large number of experts' game records of Go as training data in order for T361G to learn the evaluation function of Go board states. We reduce the number of parameters to be learned by taking Go-specific properties into account. It is shown that T361G predicts the winning probability fairly well with avoiding overtraining, even from insufficient amount of data.

Keywords: Go, neural network, supervised learning.

1 Introduction

Go is a classic board game, which has been long played in East Asia. The basic rule is simple; each of two players alternately places Black or White stone on a board, and finally the player who has larger territory on the board becomes a winner. However, it is very complex to analyze the strategy such as evaluating the significance of the stone in certain board position. This complexity has prevented making a computer Go agent that is stronger than human experts.

There are two major reasons why Go is complex to analyze; 1) The size of Go board, 19×19 , is too big and there is almost no restriction of place to put stones. Therefore, the search tree inevitably becomes large. 2) Different from Chess and Shogi pieces, Go stone alone does not have an explicit role, but the role of each stone is defined by a stone pattern whose possible combination is enormous. Because of these two reasons, the existing methods such as α - β search that was actually used for computer Chess agent [5], cannot be applicable to Go.

In recent years, however, computer Go agents have become strong remarkably due to the progress of Monte-Carlo Go [3]. The algorithm requires little knowledge of Go. It determines the next stone place by evaluating the expected winning rate that stems from the current situation by means of a large number of random simulations starting from the current situation. Furthermore, UCT algorithm, which incorporates the trade-off between 'exploration' (of the untested move) and 'exploitation' (of the best move ever tested) [1] into the

Monte-Carlo Go, was also proposed and showed the better performance than the original Monte-Carlo Go [4]. To evaluate the winning rate from the current state with satisfactory accuracy, these Monte-Carlo Go algorithms require numerous number of random simulations, which would be a hazard for real-time implementations of computer Go players.

In this study, we propose a new framework ‘T361G’ for constructing a board state evaluation function. T361G evaluates the board state using a hierarchical neural network which is trained on a large number of human experts’ game records. The hierarchical neural network receives stone positions in Black turn (or White turn) as its input, and outputs an expected winning probability of Black (or White). When training the neural network, the input stone positions are randomly chosen from the game records and the output is given either 1 or 0 depending on the actual result of the game. The expected winning probability of a given board state is regarded as the board state evaluation. Based on this evaluation function, good next move can be obtained; for example, the player’s move that yields the highest winning probability or the move that yields the highest winning probability after the opponent chooses his best possible move.

T361G has several advantages over the Monte-Carlo Go; 1) By using experts’ game records, T361G can evaluate board states obtained by more realistic moves than random simulations. 2) After the training, T361G evaluates the board state quickly because the output calculation requires a simple linear algebra and component-wise nonlinear transformation while the Monte-Carlo Go requires heavy random simulations for evaluating each move. 3) While random simulations done by the Monte-Carlo Go are just for evaluating each board state and difficult to reuse, T361G learns weight parameters of the neural network, implying T361G efficiently memorizes all the game records through training. This efficient representation of memory is expected to work well for generalization, that is, for evaluating unseen board states. On the other hand, its learning is challenging because it is difficult to learn the parameters of such a large network whose input is discrete and the dimension is as large as $19^2 = 361$ with avoiding overtraining. In this study, we overcome this difficulty by reducing adjustable parameters in the network based on Go-specific properties, and by using a stochastic gradient descent algorithm which has good scalability to the size of the training data.

This article is organized as follows; Section 2 describes the architecture of the T361G network after explaining several desirable properties for our network. The training of the T361G network is explained in Section 3. Section 4 shows results of the learning for artificial data and for human experts’ game records. Section 5 summarizes contributions of our study.

2 Architecture of T361G Network

A hierarchical neural network T361G attempts to solve a regression problem in which input \mathbf{x} represents stone positions of Go board in Black turn and its output represents a winning probability of Black;

$$p(C_B|\mathbf{x}) = f(\mathbf{x}|\boldsymbol{\theta}), \quad (1)$$

where $\boldsymbol{\theta}$ denotes all the parameters in the T361G network.

2.1 Desirable Properties for Go Neural Network

Here, we describe Go-specific properties; we will see later that these properties are fulfilled in our network.

- **Normalization of output**

The output range of our neural network should be $[0, 1]$ because it represents an expected winning probability.

- **Symmetry between Black and White**

If the expected Black's winning probability conditioned on the input board state in Black's turn is P , then the expected White's winning probability conditioned on the board state in White's turn in which the colors of all the stones on the board are reversed¹ should also be P , that is, the expected Black winning probability conditioned on the reversed board in White's turn should be $1 - P$.

- **Rotational and mirror symmetry in Go board state evaluation**

Since the relative stone positions do not change by the rotation and the mirror-image reflection of the board, the evaluation of the board state should be invariant with respect to such transformations. Because four ways of rotations and two ways of mirror-image reflections yield eight possible equivalent transformations, the output of our neural network should be consistent over such eight equivalent inputs.

2.2 Strong Local Correlations in Stone Patterns

If we employ a simple architecture in which M hidden units are connected to all the input units of $19 \times 19 = 361$, the number of weight parameters between the input and the hidden layers amount $361 \times M$. Since this number is quite large, there is a risk of overtraining and the calculation of the network output requires a large computation cost. For dimension reduction of parameters, we focus on local patterns of stones. Because it is said that there are good or bad local stone patterns, such local patterns should be paid attention to when evaluating the board state. Therefore, each hidden unit is assumed to have its receptive field on the input board, in particular, 4×4 patch on the Go board; there are no connections from other positions on the board. Furthermore, it is assumed the function of receptive fields is the same between different hidden units; the weights for connections between a hidden unit and its corresponding input patch are common over the hidden units. Such local patches (receptive fields) are made as many as possible (in total $16 \times 16 = 256$) with overlapping each other so that the patch boundary does not affect so much. Then, the number of weight parameters we should tune reduces to $16 \times m$ where m is the number of hidden units connecting local patches.

¹ Note that Black and White's symmetry does not mean there is no asymmetry for the winning probability if one takes the initial move or the second move.

2.3 Architecture of T361G Network

To incorporate the above properties, our hierarchical neural network is organized as follows. Let $\mathbf{x}^{(1)}$ be a $19 \times 19 = 361$ dimension vector representing an input board state. The i -th element x_i ($i = 1, \dots, 361$) takes 1, -1 and 0 when there is Black stone, White stone, and no stone, respectively, at the i -th position of the board. From given input $\mathbf{x}^{(1)}$, we replicate the other seven equivalent (rotational and mirror-image symmetric) board states denoted as $\mathbf{x}^{(k)}$ ($k = 2, \dots, 8$). These equivalent inputs $\mathbf{x}^{(k)}$ ($k = 1, \dots, 8$) are fed into eight identical networks to satisfy the rotational and mirror-image symmetry of the board (their integration will be seen later). Since the eight networks are identical, hereafter we explain the architecture of one of the eight networks, in which the input is simply denoted as \mathbf{x} .

First, 16-dimensional patch vectors $\mathbf{z}^{(i)}$ ($i = 1, \dots, 256$) are extracted from the input \mathbf{x} , each of which corresponds a certain 4×4 local patch on the Go board. Each patch $\mathbf{z}^{(i)}$ connects eight first-layer hidden units $h_{l,i}^{(1)}$ ($l = 1, \dots, 8$), which are obtained as

$$h_{l,i}^{(1)}(\mathbf{x}) = 2\sigma\left((\mathbf{w}_l^{(1)})^T \mathbf{z}_i\right) - 1 \quad (2)$$

where $\mathbf{w}_l^{(1)}$ ($l = 1, \dots, 8$) is a 16-dimensional weight vector and $\sigma(x) \equiv \frac{1}{1+\exp(-x)}$ denotes a logistic sigmoid function. T denotes a vector transpose. Note that $h_{l,i}^{(1)}$ extracts the same feature as far as the input local patch is same irrelevant to the position of the patch on the Go board. By representing the concatenation of 256×8 first-layer hidden units $h_{l,i}^{(1)}$ as $\mathbf{h}^{(1)}$, the second-layer hidden unit $h_m^{(2)}$ ($m = 1, \dots, 16$) is calculated as

$$h_m^{(2)}(\mathbf{x}) = 2\sigma\left((\mathbf{w}_m^{(2)})^T \mathbf{h}^{(1)}\right) - 1 \quad (3)$$

where $\mathbf{w}_m^{(2)}$ ($m = 1, \dots, 16$) is a 2048-dimensional weight vector. Finally, the third-layer hidden unit $h^{(3)}$ is calculated as

$$h^{(3)}(\mathbf{x}) = 2\sigma\left((\mathbf{w}^{(3)})^T \mathbf{h}^{(2)}\right) - 1 \quad (4)$$

where $\mathbf{w}^{(3)}$ is a 16-dimensional weight vector. Note that all the nonlinear activation functions for $h^{(1)}$, $h^{(2)}$ and $h^{(3)}$ are of the form $2\sigma(\cdot) - 1$ without additional bias term so that the signs of the output are easily flipped when color of all the stones on the Go board are reversed. Moreover, this nonlinear function is suitable for representing the ambiguity of the judgment which player will win in the early stage of the game, because the nonlinear function takes (nearly) zero when the input is (nearly) zero, i.e., few stones are placed on the Go board (note that the x_i takes zero when the i -th position of the board is vacant). Finally, the output of the T361G network, y , is obtained as the nonlinear transformation of the summation of the outputs of eight identical networks;

$$y = \sigma\left(\sum_{k=1}^8 (\mathbf{w}^{(4)})^T h^{(3)}(\mathbf{x}^{(k)})\right) \quad (5)$$

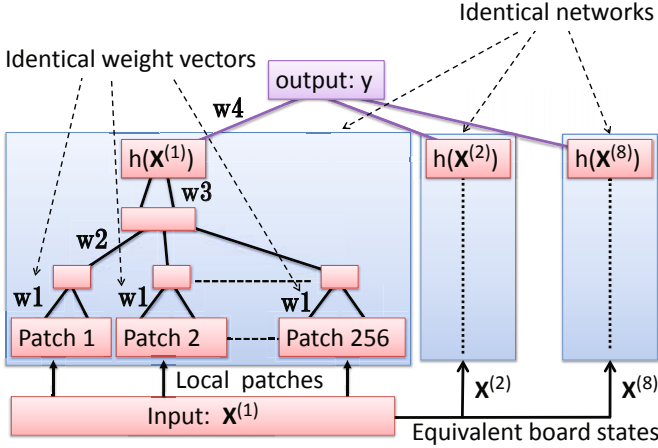


Fig. 1. The neural network architecture

so that it satisfies the Black-White symmetry and the rotational and mirror-image symmetry of the board. Here, $w^{(4)}$ is a scalar weight parameter. The total number of the parameters to be learned is 32,913. The architecture of the T361G network is depicted in Fig. 1

3 Learning Algorithm

Based on a large number of experts' game records as the training data, we performed maximum likelihood estimation of the network parameters. Let \mathbf{x}_n ($n = 1, \dots, N$) be an n -th training datum, which denotes a board state of a certain experts game in Black's turn (White's turn), and t_n be the outcome of the game where $t_n = 1$ and $t_n = 0$ denote Black's win (White's lose) and White's win (Black's lose), respectively. Then, the likelihood function becomes

$$p(t_n | \mathbf{x}_n, \theta) = y_n^{t_n} \{1 - y_n\}^{1-t_n}. \tag{6}$$

By taking negative logarithm of the likelihood function, we obtain the cross entropy error function $E_n(\theta)$ as follows.

$$E_n(\theta) \equiv -\ln p(t_n | \mathbf{x}_n, \theta) = -\{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}. \tag{7}$$

The cross entropy error function is minimized by a stochastic gradient descent method;

$$\theta^{(\text{new})} = \theta^{(\text{old})} + \eta_n \nabla E_n(\theta^{(\text{old})}), \tag{8}$$

where η_n is a learning coefficient and needs to satisfy the following conditions for convergence [6]: $\sum_{n=1}^{\infty} \eta_n = \infty$ and $\sum_{n=1}^{\infty} \eta_n^2 < \infty$. In this study, we set

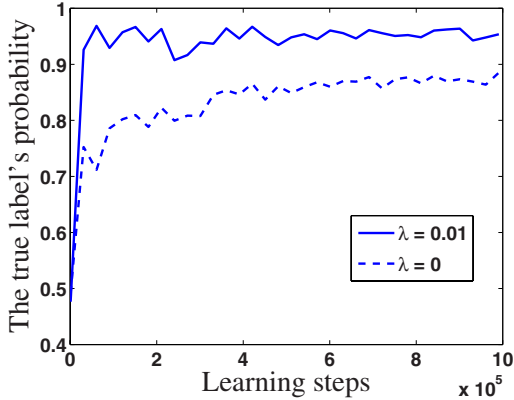


Fig. 2. Learning curve for the artificial data. The ordinate denotes the exponential of 100 test sample average of the cross entropy while the abscissa denotes the number of learning steps, i.e., the number of training data used for the learning. The learning constants are as follows: $\lambda = 0.01$, $\tau = 100,000$, and $\eta_0 = 0.1$ for the solid line while $\lambda = 0$, $\tau = 10,000$, and $\eta_0 = 0.5$ for the dashed line.

$$\eta_n = \frac{\tau}{\tau + n} \eta_0 \quad (9)$$

where η_0 and τ are positive constants. Although the maximum likelihood estimation is known to be asymptotically efficient [2], it can overfit the training data as long as the number of training samples is finite. Then, we introduce the following regularization term to the parameter update;

$$\boldsymbol{\theta}^{(\text{new})} = \boldsymbol{\theta}^{(\text{old})} - \eta_n \nabla E_n(\boldsymbol{\theta}^{(\text{old})}) - \lambda \boldsymbol{\theta}^{(\text{old})}, \quad (10)$$

where λ is a regularization coefficient whose range is $[0, 1)$. Because the regularization term decreases the absolute value of the parameter updates, it works to stabilize the learning even from a limited number of training data.

4 Experiments

4.1 Learning with Artificial Data

We examined the performance of our hierarchical neural network using an artificial dataset whose input dimension is as high as the Go board state, i.e., $19^2 = 361$. In the case of real experts' game records, the outcome could be noisy because it depends on the subsequent playing by humans, whereas we can control the noise easily in the case of artificial data. Then, we examined how many training data we need to train the T361G network with avoiding overfitting, and how the regularization term works. For each artificial datum, the input consists of 361 variables whose each value was set as 0, 1 and -1 with the probabilities 0.5, 0.25, and 0.25, respectively. The corresponding output took 1

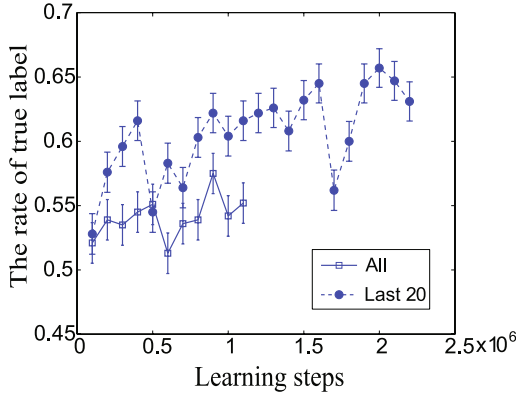


Fig. 3. Learning curve for the real experts' game records. The ordinate denotes the rate of 1000 test samples whose actual outcomes were correctly predicted by the T361G network while the abscissa denotes the number of learning steps, i.e., the number of training data used for the learning. The solid line and dashed line denote the cases where all the game states and the last 20 moves were used for the training and test, respectively.

when the sum of all the input variables was more than 0, otherwise took 0. We trained the T361G network using 1,000,000 training data, and tested on 100 test data, which were independently generated from the training data by the same method with that of the training data. To evaluate the generalization ability of the T361G network after the learning, we estimated the exponential of the test sample average of the cross entropy, which corresponds to the geometrical mean of the accuracy of the current label estimation over the test samples. The result is shown in Fig. 2.

As seen in the figure, the T361G network successfully learns the parameters of as many as 32,913. The geometrical mean of the accuracy reaches more than 95% with about 600,000 training data. Furthermore, it can be seen that an appropriate regularization term accelerates the learning.

4.2 Learning with Experts' Game Records

Real experts' game records were then used to train the T361G network. All the game records were taken from FLYGO (<http://www.flygo.net/>) of which 8,000 game records were used for training while 1,085 game records for test. Each game record has about 100 - 200 moves. The evaluation of the Go board state is difficult especially in the early stage because there are numerous number of possible progresses. To see the learnability according to the progress of the game, we tested two kinds of learning; one used all the board states for training and test, while the other used only the board states which have less than 20 moves before the game ends for training and test. In every step of the stochastic gradient descent, a training datum (a board state) was randomly chosen from a randomly

chosen game record in the dataset. In the training, it took about 210 seconds for 10,000 learning steps by Intel Dual-Core Xeon 5470 3.33 GHz. Note that T361G can evaluate the board states very quickly after the training is completed. We evaluated the generalization ability of the T361G network by the rate of test samples whose actual outcomes were correctly predicted by the T361G network. The outcome prediction was done as follows; if the output of the T361G network was more than 0.5 conditioned on the Black's turn, T361G is assumed to predict the Black's win, otherwise the White's win. As seen in Fig. 3, in the case of training and test from the last 20 moves, T361G successfully predicted the game outcomes as high as 65% for unseen board states. On the other hand, T361G predicted the game outcomes around 55% when all the Go board states were used for training and test. This performance is significantly higher than the chance level, suggesting the learnability of the Go board state evaluation function to some extent, nevertheless there could be tons of possibilities in the subsequent game progress.

5 Conclusion

When the input variables take discrete values and their dimension is high, we need a huge amount of training data, which is said "curse of dimensionality". If the number of available data is not sufficient, the machine learner like an artificial neural network should suffer from overtraining. In this study, we avoid such a difficulty by reducing adjustable parameters of the network and using a stochastic gradient descent method with a regularization. The reduction of the parameters are expected not to deteriorate the representation power of the network because the reduction is reasonably performed utilizing Go-specific properties (symmetry and locality explained in Sections 2.2 and 2.3), and it was justified from the real data learning, showing successful prediction of the winning probability. Although we assumed identical weight parameters between each of hidden units and corresponding local patch inputs irrelevant to the location of the patch on the Go board, we can extend our design into incorporating special receptive field for 'kado' and 'hen', which are clearly different from other inside spaces. Such an extension will be done as a near future study.

References

1. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47, 235–256 (2002)
2. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer, New York (2006)
3. Bruegmann, B.: Monte Carlo Go, <ftp://ftp-igs.joyjoy.net/go/computer/mcgo.tex.z>
4. Kocsis, L., Szepesvari, C.: Bandit based Monte-Carlo planning. In: 15th European Conference on Machine Learning, pp. 282–293 (2006)
5. Newborn, M.: *Computer Chess Comes of Age*. Springer, Heidelberg (1996)
6. Robbins, H., Monro, S.: A stochastic approximation method. *Annals of Mathematical Statistics* 22, 400–407 (1951); Springer-Verlag (1996)

Quick Maximum Power Point Tracking of Photovoltaic Using Online Learning Neural Network

Yasushi Kohata¹, Koichiro Yamauchi², and Masahito Kurihara¹

¹ Graduate School of Information Science and Technology, Hokkaido University, Sapporo Hokkaido 060-0814, Japan

² Chubu University, Department of Information Science, 1200, Matsumoto-cho, Kasugai-shi, Aichi 487-8501, Japan

Abstract. It is well known that the photovoltaic (PV) device has a Maximum Power Point (MPP) that can ensure that maximum power is generated in a device. Since this MPP depends on solar radiation and the PV-panel temperature, it is never constant over time. A Maximum Power Point Tracker (MPPT) is widely used to ensure there is maximum power at all times. Almost all MPPT systems use a Perturbation and Observation (P&O) method because its simple procedure. If the solar radiation rapidly changes, however, the P&O efficiency degrades.

We propose a novel MPPT system to solve this problem that covers both the online-learning of the PV-properties and the feed-forward control of the DC-DC converter with a neural network. Both the simulation results and the actual device behaviors of our proposed MPPT method performed very efficiently even when the solar radiation rapidly changed.

Keywords: Photovoltaic, MPPT, P&O, Online Learning, non-i.i.d. data.

1 Introduction

A photovoltaic (PV) device is a type of current source, whose properties vary depending on the level of solar radiation and the PV panel temperature. For example, if there is a lot of solar radiation, the PV generates a large current, but the current is reduced if the PV voltage is larger than a given voltage. The properties are also valid depending on the temperature. Therefore, PV has a Maximum Power Point (MPP) that can ensure that maximum power is generated. To ensure there is maximum power at all times, a Maximum Power Point Tracker (MPPT) is needed. Almost all MPPT systems use a Perturbation and Observation (P&O) method because of its simple procedure. However, if the level of solar radiation rapidly changes, the efficiency of P&O degrades.

Several MPPT controllers that use a Neural Network (NN) have been proposed [1] [2] [3] to solve this problem. Although these MPPT controllers quickly respond to the rapidly changing solar radiation, almost all of them need to do pre-learning using PV specific data.

We propose a novel MPPT system to solve this problem that covers both the online-learning of the PV-properties and the feed-forward control of the DC-DC converter simultaneously. The approach of the proposed method is a combination of an online learning Neural Network and the P&O method. Using this approach, the system does not need any prior exploitation to adjust to the PV specifications. To do this, a general regression neural network [4] [5] is used to achieve the online learning of non-stationary inputs (non-i.i.d. inputs).

2 Photovoltaic and MPPT

Figure 1 shows an example of the behavior of a PV device. The magnitude of the generated current varies depending on the solar radiation S , the PV temperature T , and the voltage V_{PV} (Fig. 1).

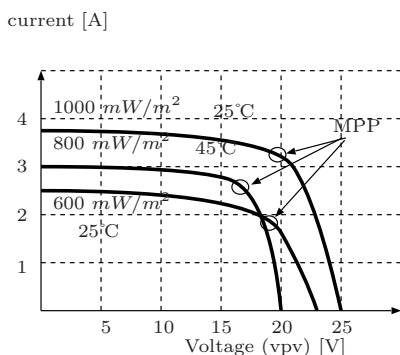


Fig. 1. Example of PV properties

Therefore, there is a maximum power point (MPP) for each (S, T) . Note that the relation between the MPPs and (S, T) s is a nonlinear PV-panel specific function.

To find the MPP, a perturbation and observation (P&O) method, which sequentially optimizes the PV-voltage in order to maximize the generated power, is widely used. The P&O monitors whether the generated power is increased or not due to the change in voltage ΔV . If the power is increased, it decides that the next change in voltage should be the same as that of the last one (ΔV). However, if the power is decreased, it decides that the next change in voltage should be a negative one $-\Delta V$.

Although the P&O method is a very simple procedure to achieve the MPPT, it cannot track the MPP very well if the solar radiation rapidly changes. To overcome this drawback, we propose a novel MPPT method that uses a neural network, which achieves the learning and maximum power point tracking simultaneously.

3 Proposed System

3.1 Outline

The proposed method consists of the Neural Network (NN) and the P&O parts (Fig. 2, 3). When solar radiation only slowly changes, the system controls the DC-DC converter by using the P&O and, thus, the NN can learn the MPP, which is founded by the P&O, simultaneously. On the other hand, when the solar radiation is rapidly changing, the system controls the DC-DC converter by using the NN so that it can track the MPP without delay.

Normally, neural networks need independent and identically distributed (i.i.d.) samples to ensure successful online learning. In this case, however, similar training samples, which depend on the angle of incident radiation, will be consecutively given to the NN. To deal with these training samples, we use a General Regression Neural Network (GRNN) [4] for the proposed system to accomplish the stable learning.

3.2 Neural Network Used

In this system, the neural network has to learn each sample in an online manner, because it is difficult to store all the learning samples in the small device. Moreover, the sample distribution is not i.i.d., but the prior distribution of inputs $P(x)$, where $x = (S, T)^T$, varies depending on the angle of incident radiation.

The general regression neural network (GRNN) [4] ensures stable learning in such environments. GRNN is a memory-based network that provides the estimates of the continuous variables. Even with sparse data in a multidimensional measurement space, the algorithm provides smooth transitions from one observed value to another. GRNN consists of an input layer, a pattern layer, a summation layer, and an output layer. The output $y(x)$ of GRNN is

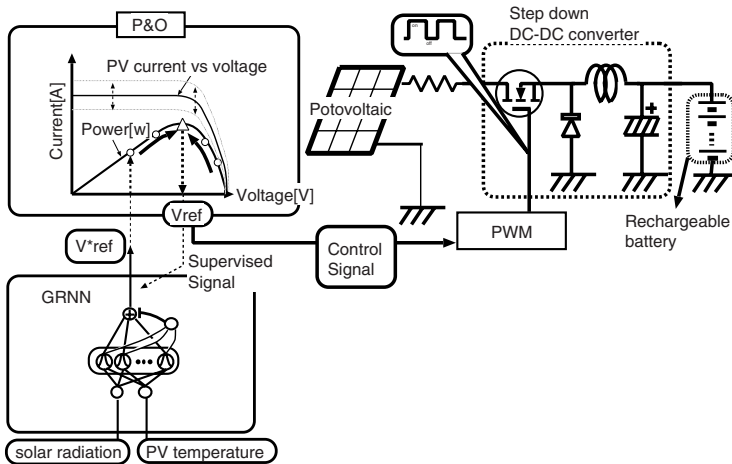


Fig. 2. Outline of system

$$y(\mathbf{x}) = \frac{\sum_{j=1}^J w_j \phi_j(\mathbf{x})}{\sum_{j=1}^J \phi_j(\mathbf{x})}, \quad \text{where } \phi_j(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_j\|^2}{2\sigma^2}\right), \quad (1)$$

where \mathbf{x} is the input vector, J is the number of pattern units, ϕ_j is the basis function of pattern unit j , \mathbf{c}_j is the center of the basis function, and w_j is the weight. σ^2 denotes the variance of the basis function.

The learning by GRNN is achieved by adding a new unit ($J + 1$ -th unit) to the pattern layer, and its new weight w_{J+1} is set to the target output \hat{y} of the training sample. In this study, if the given training sample is similar to one of the already learned samples, GRNN does not add the new unit, but updates the weight of the unit that is the nearest to the training sample. This is to avoid wasting not only the memory capacity but also the computational power of the system.

Therefore, when the training sample $[\hat{\mathbf{x}}, \hat{y}]$ is given, the GRNN calculates $d_{nearest} = \|\mathbf{x} - \mathbf{c}_{nearest}\|^2$, which is the nearest center to $\hat{\mathbf{x}}$. Then, if $d_{nearest} > \theta$, the GRNN adds a new unit whose parameters are initialized as

$$w_{new} = \hat{y}, \quad \mathbf{c}_{new} = \hat{\mathbf{x}}. \quad (2)$$

On the other hand, when $d_{nearest} < \theta$, it updates the weight of the nearest unit. The weight is modified so that it is close to \hat{y} according to the following equation,

$$w_{nearest} = (nw_{nearest} + \hat{y}) / (n + 1), \quad (3)$$

where n is the number of samples given to the unit. Note that the above equation means that $w_{nearest}$ is set to the averaged target values, which are given to the unit. Using this strategy, the GRNN reduces the wrong influence due to noisy target values. In this learning algorithm, if θ is large, the number of units decreases, but the mean square error (MSE) of the GRNN will be large. On the other hand, if θ is small, the MSE may decrease, but the number of units increases. Therefore, we need to define θ according to the available memory capacity and desired accuracy. Moreover, σ should also be defined according to the accuracy of the resultant network. We determined σ using numerical tests.

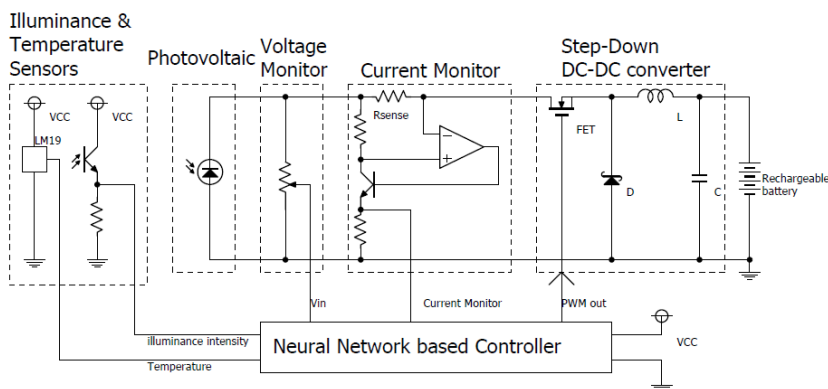


Fig. 3. Circuit configuration

Although the learning method is based on the GRNN, it is similar to that of ART-map [6]. However, this learning algorithm is simpler than that of ART-map.

3.3 Whole Algorithm

Algorithm 1 lists the pseudo code of the algorithm for the proposed MPPT controller. Note that the controller repeats Algorithm 1.

If the change in solar radiation ΔS is less than a threshold γ , the system finds the best control signal V_{ref} for the observed input $\mathbf{x} = (S, T)^T$ using the P&O method. Then, the GRNN learns (\mathbf{x}, V_{ref}) incrementally. On the other hand, if $\Delta S \geq \gamma$, the system uses the GRNN output $y(\mathbf{x})$ as V_{ref} .

If the threshold γ is too large, the GRNN is not used, so the proposed system is equivalent to P&O. In this study, we set γ to the mean value of the noise of the output from the solar radiation sensor.

If the input is far from the learned samples, however, the GRNN output is untrustworthy. To avoid using such GRNN outputs, the system does not use the GRNN output when $\|\mathbf{x} - \mathbf{c}_{nearest}\|^2 \geq \theta$.

Algorithm 1. Pseudo code of proposed MPPT algorithm

```

Measure solar radiation  $S$ , temperature  $T$ , PV voltage  $V_{PV}$  and current  $I$ 
 $\mathbf{x} = (S, T)^T$ ,  $\Delta S := S - S_{previous}$ ,  $P := IV_{PV}$ 
 $\mathbf{c}_{nearest}$  = the nearest center of GRNN pattern unit to  $\mathbf{x}$ 
if  $|\Delta S| > \gamma$  and  $\|\mathbf{x} - \mathbf{c}_{nearest}\|^2 < \theta$  then
   $V_{ref}$  = GRNN output  $y$  (eq. 1)
else
  if  $P < P_{previous}$  then
     $\Delta V := -\Delta V$ 
  end if
   $V_{ref} := V_{ref} + \Delta V$ 
end if
if moving average of  $V_{ref}$  during the last 20 steps == 0 then
  Make the GRNN learn  $(\mathbf{x}, V_{ref})$ 
end if
 $S_{previous} := S$ 
 $P_{previous} := P$ 
return  $V_{ref}$ 

```

4 Experiments

4.1 Computer Simulation

We verified the proposed method by using a computer simulation.

In this simulation, the changes in solar radiation and panel temperature were made by the sine curve [4], and the noise and the response speeds of various equipment were disregarded.

¹ We assume that the solar radiation curves can be represented by the combination of several sin curves, which can be derived by using the Fourier Transformation.

First, we changed the solar radiation and panel temperature gradually according to low frequency sine curves to confirm whether or not the proposed system can learn the PV characteristics under gradually changing solar radiation conditions. At this time, the frequency of the sine curve, which represents the changes in solar radiation and panel temperature, was slightly staggered so that the proposed system could study the MPPs of various solar radiation and panel temperature combinations.

Second, we used high frequency sine curves to simulate the rapid changes in solar radiation to show the effectiveness of the proposed system under rapidly changing solar radiation conditions, and the proposed system without the neural network (P&O only) method was also tested for comparison (Fig. 4). We can see that the proposed system basically maintained about an optimum output whereas the output of the P&O was a little lower than the optimum value.

Finally, we compared the performances of the proposed system with that of the optimal system, which can completely track the MPP, under the sine curved solar radiation with varying frequencies. Here, we examined the efficiency r where $r = W_{out}/W_{best}$, W_{out} is the output power from the proposed system, and W_{best} is the optimum energy. In Figure 5 the horizontal axis is the frequency f , and the vertical axis is the efficiency r . From this figure, we can see that the efficiency of P&O decreases as the frequency increases while the proposed system yields about a 99 % efficiency at any frequency.

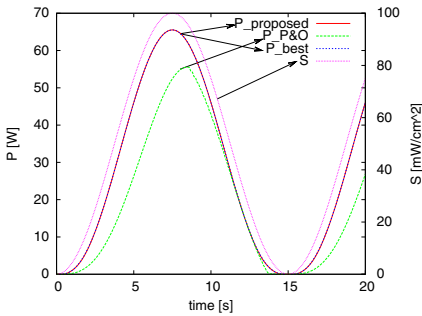


Fig. 4. Output power from each system under sine curved solar radiation, where S denotes solar radiation

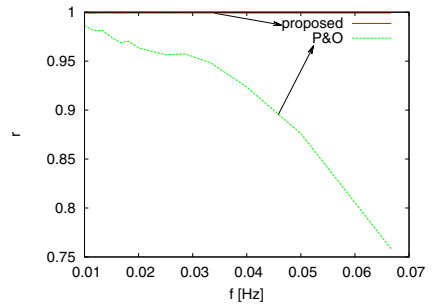


Fig. 5. Relation between frequency f and efficiency r

4.2 System Behavior of Actual Device

We also verified our proposed method using actual devices. We checked whether the proposed system achieves MPP learning and quick control when solar radiation rapidly changed.

The actual device consists of a silicon photovoltaic device of 20 V/20 W, a step down switching converter, a 7.2-V Ni-MH rechargeable battery, and a laptop computer (Fig. 3, 6). The Ni-MH rechargeable battery was connected to the photovoltaic device through the switching converter. The laptop computer

executed the MPPT algorithms and controlled the switching converter through the analog input/output board, and recorded the responses from the system sensors (Fig. 3).

In the experiment, we changed the installation angle of the PV-panel to change the apparent solar radiation. Moreover, we executed not only the proposed method but also the original P&O method for comparison. The simplest way to compare these two methods was by preparing two sets of devices for the proposed and P&O methods. However, it is very difficult to make precisely the same conditions for the different devices. To overcome this difficulty, we made the laptop PC execute the two methods alternately. Namely, each method sent a control signal to the switching converter alternately every 100 milliseconds, and the two methods were switched every 50 milliseconds.

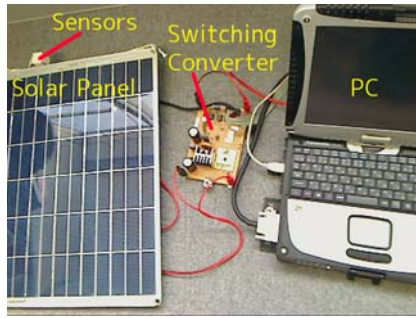


Fig. 6. Actual device: PV-panel with illuminance, temperature sensors, and PC controlled switching converter

First, we fixed the PV-panel angle horizontally, and verified the behaviors of the two methods (Fig. 7). In the graph, although the vertical axis should originally set the unit of the illuminance and temperature, the measure is [voltage] because all the sensory outputs were straightforwardly plotted. Moreover, the operation voltage V and output power P were multiplied by the coefficients (< 1), which is determined by looking at the characteristics of the sensors.

According to Figure 7, by gradually changing the solar radiation conditions, both the proposed and P&O methods approached the MPP in a similar manner, because the proposed method accomplishes the same operation as that of the original P&O under the given conditions. However, the GRNN learned the best control signals when using the proposed method.

Next, we changed the installation angle of the PV-panel, and made the proposed method work for one minute on each angle in order to let the GRNN learn the MPPs under various solar radiation conditions.

Finally, we intensely changed the installation angle of the PV-panel while the two methods were working (Fig. 8). According to Figure 8, the proposed method was able to generate greater power than that of the original P&O, because the proposed method was able to reach the MPP immediately by using the GRNN while the original P&O method gradually approaches MPP.

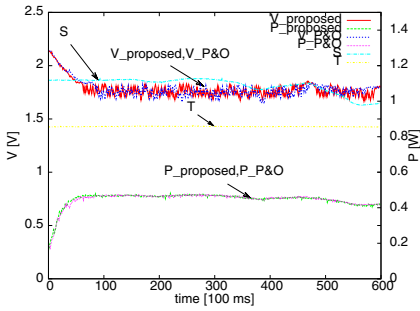


Fig. 7. Example of system behaviors under gradually changing solar radiation conditions, where S, T and P denote solar radiation, temperature and power, respectively

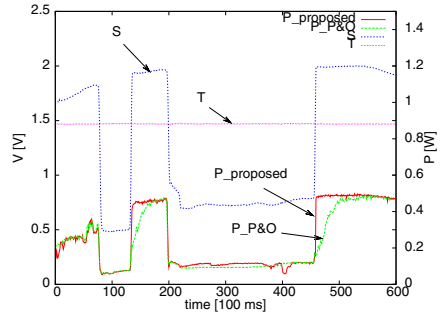


Fig. 8. Example of system behaviors under rapidly changing solar radiation conditions, where S, T and P denote solar radiation, temperature and power, respectively

5 Conclusion

In this study, we proposed a maximum power point tracking converter that uses an online learning neural network and the perturbation and observation (P&O) method. We showed that the proposed system is able to learn the photovoltaic properties while operating the P&O under gradually changing solar radiation conditions, and accomplishes the quick tracking of the maximum power point when the solar radiation is rapidly changing. We do not need any prior exploitation for adjusting to the PV specifications when using the proposed MPPT method.

The proposed system, however, cannot easily learn if the solar radiation is always rapidly changing. However, we think this is rare in actual environments.

We are planning to embed the proposed method into a micro-computer in the near future.

References

1. Hiyama, T., Kitabayashi, K.: Neural network based estimation of maximum power generation from pv module using environmental information. *IEEE Transactions on Energy Conversion* 12(3), 241–247 (1997)
2. AbdulHadi, M., Al-Ibrahim, A.M., Virk, G.S.: Neuro-fuzzy-based solar cell model. *IEEE Transactions on Energy Conversion* 19(3), 619–624 (2004)
3. Akkaya, R., Kulaksiz, A.A., Aydogdu, O.: Dsp implementation of a pv system with ga-mlp-nn based mppt controller supplying bldc motor drive. *Energy Conversion & Management* 48, 210–218 (2007)
4. Specht, D.F.: A general regression neural network. *IEEE Transactions on Neural Networks* 2(6), 568–576 (1991)
5. Tomandl, D., Schober, A.: A modified general regression neural network (mgrnn) with a new efficient training algorithm as a robust “black-box”-tool for data analysis. *Neural Networks* 14, 1023–1034 (2001)
6. Su, M.-C., Lee, J., Hsieh, K.-L.: A new ARTMAP-based neural network for incremental learning. *Neurocomputing* 69, 2284–2300 (2006)

Semi-Naïve Bayesian Method for Network Intrusion Detection System

Mrutyunjaya Panda¹ and Manas Ranjan Patra²

¹ Department of ECE, Gandhi Institute of Engineering and Technology, Gunupur,
Orissa-765022, India

mrutyunjaya.2007@rediffmail.com

² Department of Computer Science, Berhampur University-760007, Orissa, India
mrpatra12@gmail.com

Abstract. Intrusion detection can be considered as a classification task that attempts to classify a request to access network services as safe or malicious. Data mining techniques are being used to extract valuable information that can help in detecting intrusions. In this paper, we evaluate the performance of rule based classifiers like: JRip, RIDOR, NNge and Decision Table (DT) with Naïve Bayes (NB) along with their ensemble approach. We also propose to use the Semi-Naïve Bayesian approach (DTNB) that combines Naïve Bayes with the induction of Decision Tables in order to enhance the performance of an intrusion detection system. Experimental results show that the proposed approach is faster, reliable, and accurate with low false positive rates, which are the essential features of an efficient network intrusion detection system.

Keywords: Intrusion Detection, Rule Based Classifiers, Hybrid DTNB, Ensemble approach, Accuracy.

1 Introduction

With the growing use of Internet, information security threat is becoming one of the most forbidding problems. The demand for reliable connection, information integrity and privacy is more intense today than ever before. One possible precaution is the use of an effective Intrusion Detection System (IDS).

Data Mining is a relatively new approach for intrusion detection. Data mining approaches for intrusion detection was first implemented in mining audit data for building automated models for Intrusion Detection [1]. The raw data is first converted into ASCII network packet information which in turn is converted into connection level information. These connection level information records contain connection features like service, duration, protocol, etc. Data mining algorithms are applied to this data to create models to detect intrusions.

In this paper, we investigate and evaluate the performance of various rule based classifiers like JRip, Ridor, NNge, and Decision Table (DT), Bayesian classification using Naïve Bayes (NB), Hybrid DTNB and an ensemble approach. The motivation for using the hybrid approach is to improve the detection accuracy of an IDS compared to

using individual approaches. Finally, we use AdaBoost algorithm as an ensemble approach to all the above for further enhancement in the intrusion detection accuracy while maintaining low false positive rate. The rest of the paper is organized as follows. Related research is presented in Section 2 followed by a short theoretical background on the rule based classification algorithms in Section 3. A brief introduction to Naïve Bayesian classifiers is presented in Section 4. Hybrid classifiers and ensemble approach used in this research is discussed in Section 5. Experimental results and analysis is presented in Section 6 followed by conclusion in Section 7.

2 Related Research

In [2], the authors include a hybrid statistical approach which uses Data Mining and Decision tree classification in identifying the false alarms. In that, the authors conclude that their strategy can be used to evaluate and enhance the capability of an IDS to detect and at the same time to respond to the threats and benign traffic in critical network applications. . The authors in [3] present two hybrid approaches for modeling IDS. Decision trees and SVM are combined as a hierarchical hybrid intelligent system model (DT-SVM) and an ensemble approach combining the base classifiers. They conclude that the proposed research provides more accurate intrusion detection capabilities. Intrusion detection using an ensemble of intelligent paradigms is proposed in [4]. In this, the authors show that an ensemble of ANNS, SVMs and MARS is superior to individual approaches for intrusion detection in terms of classification accuracy. In [5], the authors present an intrusion detection model based on hybrid neural network and C4.5. The key idea is to take advantage of different classification capabilities of neural network and the C4.5 algorithm for different attacks. However, in this, they consider only few selected attacks from each category for their analysis. A review of various supervised classification techniques is presented in [6]. In [7], the authors propose hybrid GA (genetic algorithm) /decision tree algorithm which outperform the decision tree classifier in order to build a network intrusion detection model. In this, they conclude that this improvement is due to the fact that the hybrid approach is able to focus on relevant features and eliminate unnecessary and distracting features. However, the hybrid GA /decision tree algorithm needs to be tested more in depth for its true potential. The authors propose a double multiple-model approach capable of enhancing the overall performance of IDS in [8]. In that, the authors adopted three reasoning methods: Naïve Bayesian, Neural Nets, and Decision Trees for IDS model. Finally, the authors conclude that even if a given model outperforms others in specific problem, it is incapable of producing better results in general. This is specifically true in case of intrusion detection because often single algorithm can't deal with all attack classes at the desired accuracy level. Thus, combination of multiple models tries to take advantage of the characteristics of the individual base models to improve overall performance of an IDS.

3 Rule Based Classifiers

In this section, we will focus on some very important and yet novel rule based classification algorithms like NNge, JRip, RIDOR, Decision table(DT), which are not yet explored by intrusion detection researchers to the best of our knowledge.

3.1 NNge (Non-Nested Generalized Exemplars)

NNge is a novel algorithm that generalizes exemplars without nesting or overlap. NNge is an extension of Nge [9], which performs generalization by merging exemplars, forming hyperrectangles in feature space that represent conjunctive rules with internal disjunction. NNge forms a generalization each time a new example is added to the database, by joining it to its nearest neighbor of the same class. Details about this algorithm can be found in [10].

3.2 JRip (Extended Repeated Incremental Pruning)

JRip implements a propositional rule learner, “Repeated Incremental Pruning to Produce Error Reduction” (RIPPER), as proposed in [11]. JRip is a rule learner alike in principle to the commercial rule learner RIPPER. RIPPER rule learning algorithm is an extended version of learning algorithm IREP (Incremental Reduced Error Pruning). Initially, a set of training examples is partitioned into two subsets, a growing set and a pruning set. The rule set begins with an empty rule set and rules are added incrementally until no negative examples are covered. This approach performs efficiently on large and noisy datasets.

3.3 RIDOR (Ripple-Down Rules)

RIDOR generates the default rule first and then the exceptions for the default rule with the least (weighted) error rate. Later, it generates the best exception rules for each exception and iterates until no exceptions are left. It performs a tree-like expansion of exceptions and the leaves have only default rules but no exceptions. The exceptions are a set of rules that predict the improper instances in default rules [12].

3.4 Decision Tables

Decision Tables are one of the possible simplest hypothesis spaces, and usually they are easy to understand. A decision table is an organizational or programming tool for the representation of discrete functions. It can be viewed as a matrix where the upper rows specify sets of conditions and the lower ones indicate sets of actions to be taken when the corresponding conditions are satisfied; thus each column, called a rule, describes a procedure of the type “if conditions, then actions”. Details about the rule based classifiers can be found in [13].

4 Naïve Bayesian Approach

The Naïve Bayes model is a heavily simplified Bayesian probability model [14]. Here, one considers the probability of an end result given several related evidence variables. The probability of end result is encoded in the model along with the probability of the evidence variables occurring, given that the end result occurs. The probability of an

evidence variable given that the end result occurs is assumed to be independent of the probability of other evidence variables given that end results occur. In [15], the authors examine the circumstances under which the Naïve Bayes classifier performs well and why. They state that the error is a result of three factors: training data noise, bias, and variance. Training data noise can only be minimized by choosing good training data. The training data must be divided into various groups by the machine learning algorithms. Bias is the error due to groupings in the training data being very large. Variance is the error due to those groupings being too small.

5 Proposed Methodology

5.1 Hybrid DTNB: A Semi-Naïve Bayesian Approach

Here we, explore the effectiveness of the simple semi-Naïve Bayesian ranking method that combines Naïve Bayes (NB) with induction of Decision Tables (DT), which is called as hybrid DTNB. This algorithm is recently proposed in [16], which to the best of our knowledge has not been used by any of the intrusion detection researchers. In this model, Naïve Bayes and Decision tables can both be trained efficiently, and the same holds true for the combined semi-Naïve Bayes model. Figure 1 shows the architecture of the semi-Naïve Bayesian approach by combining DT with NB.

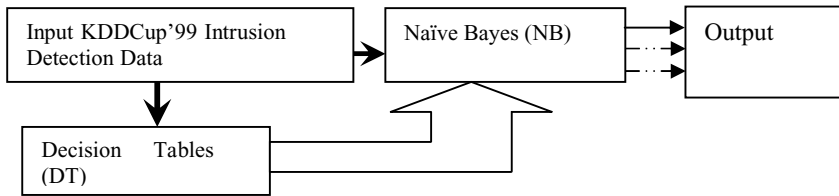


Fig. 1. Semi-Naïve Bayesian Approach

Algorithm Description. The algorithm for learning the combined model (DTNB) proceeds in much the same way as the DTs alone. At each point in the search; it evaluates the merit associated with splitting the attributes into two disjoint subsets: one for the Naïve Bayes and the other for the Decision Tables. In this, forward selection is used, where at each step, selected attributes are modeled by NB and the remainder by the DT and all attributes are modeled by the DT initially. We use leave-one-out cross validation to evaluate the quality of a split based on the probability estimates generated by the combined model. In [16], the authors use the AUC (Area under the curve) as the performance measures for the evaluation of classifiers in 2-class classification problem, whereas we aim to use accuracy as our performance measures in a 5-class classification process in building a network intrusion detection system. The class probability estimates of the Naïve Bayes and Decision Tables must be combined to generate overall class probability estimates. All probabilities are estimated using Laplace corrected

observed counts. In addition to this, a variant that includes attribute selection, which can discard attributes entirely from the combined model, is considered. To achieve this, an attribute can be discarded rather than added to the NB model, in each step of the forward selection.

5.2 Ensemble Approach

Here the idea is to apply an ensemble approach which basically does not rely on a single best classifier for decision on an intrusion; instead information from different individual classifiers is combined to take the final decision. However, the effectiveness of the ensemble approach depends on the accuracy and diversity of the base classifiers used. The architecture of the proposed ensemble approach is shown in Figure 2.

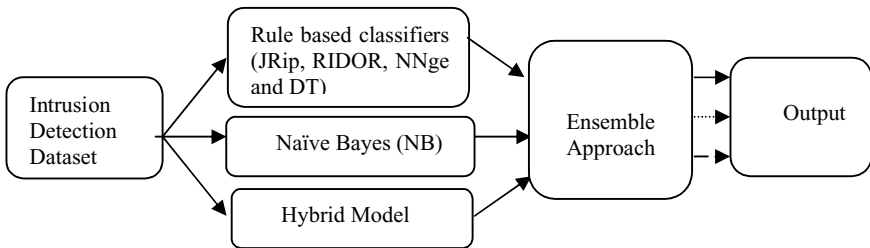


Fig. 2. Ensemble Approach

6 Experimental Results and Discussion

We use KDDCup 1999 intrusion detection benchmark dataset for our experiments. The data set contains 24 attacks and 41 attributes. We have randomly selected 1000 connection records out of those, which contains all intrusion types, where the care has been taken to include all the rare attacks that fall under U2R and R2L category. We use five class classifications for our experimentation in building a network intrusion detection system. Full dataset is used as training data in order to build an intrusion detection system, while 10-fold cross validation is used in order to find the efficacy of the model built in the training phase. All our experiments are carried out on a Pentium 4 IBM PC with 2.8GHz CPU, 40GB HDD and 512 MB RAM.

6.1 Comparison of Results

It can be observed from Table 1 that DTNB approach enhances the detection rate of Naïve Bayes classifier in detecting Normal, Probe and U2R attack, where as it fails to perform well in case of DoS and R2L attacks. It is also observed that DTNB does not perform well for our intrusion detection dataset in comparison to NNge rule based classifiers. So, we need to use ensemble approach for Semi naïve Bayesian method in

order to build an efficient intrusion detection system, which is shown in Table 2. It can be observed from Table 2 that the performance of Hybrid DTNB is enhanced after using the ensemble approaches. It is also quite clear that the ensembled DTNB produces better detection rate in all five class than the individual DT and NB approaches. However, still, it produces low detection rate in case of rare attacks in comparison to NNge rule based classifier. Low Root mean square error (RMSE) and high kappa value makes our proposed approach more interesting in designing a network intrusion detection system. Other performance measures are also presented in Table 2 in order to provide a comparative view of the performance of each of the classifiers under consideration.

Table 1. Performance Comparison of Classifiers

		JRip	RIDOR	NNge	DT	NB	Hybrid DTNB
DR	Normal	0.9835	0.9859	0.9835	0.9859	0.96	0.979
	Probe	0.5625	0.75	0.6562	0.4375	0.279	0.406
	DoS	0.998	1.0	1.0	0.9684	0.984	0.972
	U2R	0.25	0.0	0.75	0.4444	0.0	0.5
	R2L	0.353	0.4706	0.647	0.353	0.353	0.294
RR	Normal	0.9698	0.979	0.9721	0.9188	0.927	0.941
	Probe	0.9	0.75	0.7241	0.8235	0.414	0.684
	DoS	0.9656	0.99	0.998	0.9723	0.958	0.974
	U2R	0.8333	0.8333	0.7273	1.0	0.0	1.0
	R2L	0.6666	0.6666	0.9166	0.75	1.0	0.263
FPR	Normal	0.0126	0.0107	0.0126	0.0124	0.031	0.017
	Probe	0.0142	8.35x10⁻³	0.0114	0.0182	0.032	0.019
	DoS	2.23x10 ⁻³	0.0	0.0	0.033	0.017	0.0288
	U2R	4x10 ⁻³	4x10 ⁻³	1.02x10⁻³	5x10 ⁻³	1.0	0.005
	R2L	0.0111	9.15x10 ⁻³	6.1x10 ⁻³	0.011	0.011	0.001
FNR	Normal	0.0302	0.021	0.0279	0.081	0.073	0.058
	Probe	0.1	0.25	0.2759	0.1765	0.586	0.31
	DoS	0.0343	9.76x10 ⁻³	1.97x10⁻³	0.0277	0.042	0.026
	U2R	0.1666	0.1666	0.2727	0.0	0.0	0.0
	R2L	0.3333	0.3333	0.0833	0.25	0.0	0.73
F-Value	Normal	0.9766	0.9824	0.9777	0.9512	0.943	0.96
	Probe	0.6923	0.75	0.6885	0.5714	0.333	0.51
	DoS	0.9815	0.9951	0.999	0.9703	0.971	0.973
	U2R	0.6666	0.6666	0.8	0.615	0.0	0.615
	R2L	0.4616	0.5517	0.7586	0.48	0.522	0.278
Kappa		0.9308	0.9477	0.9495	0.8915	0.87	0.887
Time Taken in Seconds		0.49	0.72	0.23	0.34	0.08	0.92
RMSE		0.0601	0.0537	0.0546	0.0919	0.072	0.0855

Table 2. Ensemble Approach

		Ensemble Approach with Base Classifiers					
		JRip	RIDOR	NNge	DT	NB	DTNB
DR	Normal	0.9929	0.986	0.9812	0.993	0.957	0.995
	Probe	0.6875	0.375	0.6875	0.75	0.791	0.854
	DoS	0.998	1.0	0.998	1.0	1.0	1.0
	U2R	0.75	0.5	0.75	0.5	0.5	0.6
	R2L	0.353	0.412	0.647	0.42	0.412	0.47
RR	Normal	0.9635	0.9813	0.972	0.979	0.958	0.972
	Probe	0.88	0.8	0.7333	0.857	0.654	0.921
	DoS	0.9883	0.9512	0.996	0.9826	0.998	1.0
	U2R	0.8571	0.8	0.8	1.0	0.667	0.67
	R2L	0.75	0.7	0.846	0.78	0.875	0.888
FPR	Normal	5.56×10^{-3}	0.01	0.014	5.52×10^{-3}	0.031	3.6×10^{-3}
	Probe	0.01	0.02	0.01	8.28×10^{-3}	9.62×10^{-3}	6.2×10^{-3}
	DoS	2.09×10^{-3}	0.0	2.08×10^{-3}	0.0	0.0	0.0
	U2R	3.04×10^{-3}	5×10^{-3}	1.02×10^{-3}	3.03×10^{-3}	2×10^{-3}	2.03×10^{-3}
	R2L	0.011	0.01	6.11×10^{-3}	0.01	0.01	9.1×10^{-3}
FNR	Normal	0.0365	0.0187	0.028	0.021	0.042	0.027
	Probe	0.12	0.2	0.0266	0.143	0.346	0.079
	DoS	0.0117	0.0487	3.94×10^{-3}	0.017	1.97×10^{-3}	0.0
	U2R	0.1428	0.2	0.2	0.0	0.333	0.333
	R2L	0.25	0.3	0.154	0.222	0.125	0.111
F-Value	Normal	0.978	0.984	0.9766	0.986	0.957	0.994
	Probe	0.772	0.51	0.71	0.8	0.716	0.886
	DoS	0.993	0.975	0.997	0.9912	0.999	1.0
	U2R	0.75	0.572	0.842	0.803	0.571	0.633
	R2L	0.48	0.52	0.733	0.546	0.56	0.625
Kappa		0.947	0.9477	0.9442	0.9489	0.921	0.958
Time Taken (Sec.)		2.66	3.28	0.94	1.53	0.59	10.6
RMSE		0.0502	0.0482	0.0512	0.0481	0.058	0.0451

7 Conclusion

In this research, we have investigated some new techniques for network intrusion detection and evaluated their performance based on the KDDCup 1999 benchmark intrusion detection dataset. We have explored rule based classifiers and Naïve Bayes as intrusion detection models. Next, we designed a semi-Naïve Bayesian approach Hybrid DTNB by combining Decision Table (DT) and Naïve Bayes (NB) and an ensemble approach with all the rule based classifiers and Hybrid DTNB as base classifier. The experimental results reveal that the proposed ensemble approach for semi-Naïve Bayesian classification performs well for Normal, Probe and Dos attacks. In Normal and DoS attacks, the detection rate is almost 100%. This result suggests that by choosing proper base classifiers 100% accuracy might be possible for other classes too.

References

- [1] MIT Lincoln Laboratory, <http://www.ll.mit.edu/IST/ideval/>
- [2] Annur, N.B., Sallehudin, H., Gani, A., Zakari, O.: Identifying false alarm for network intrusion detection system using hybrid data mining and decision tree. *Malaysian journal of computer science* 21(2), 101–115 (2008)
- [3] Peddabachigari, S., Abraham, A., Grosan, C., Thomas, J.: Modelling IDS using hybrid intelligent systems. *Journal of network and computer applications* 30(1), 114–132 (2007)
- [4] Mukkamala, S., Sung, A.H., Abraham, A.: Intrusion detection using an ensemble of intelligent paradigms. *Journal of network and computer applications* 28(2005), 167–182 (2005)
- [5] Pan, Z.-S., Chen, S.-C., Hu, G.-B., Zhang, D.-Q.: Hybrid neural network and C4.5 for Misuse detection. In: *Proc. of International conference on Machine Learning and Cybernetics*, Xi'an, November 2-5, pp. 2463–2467. IEEE Press, USA (2003)
- [6] Kotsiantis, S.B.: Supervised machine learning: A review of classification Techniques. *Informatica* 31, 249–268 (2007)
- [7] Stein, G., Chen, B., Wu, A.S., Hua, K.A.: Decision Tree classifier for network intrusion detection with GA-based feature selection. In: *Proc. of the 43rd Annual South East Regional Conference*, kennesa, Georgia, vol. 12, pp. 136–141 (2005)
- [8] Katar, C.: Combining multiple techniques for intrusion detection. *Intl. Journal of Comp.Sc and Net.Security (IJCSNS)* 6(2B), 208–218 (2006)
- [9] Salzberg, S.: A nearest hyperrectangle learning method. *Machine learning* 6, 277–309 (1991)
- [10] Roy, S.: Nearest Neighbour with generalization, Christchurch, NZ (2002)
- [11] Cohen, W.W.: Fast effective rule induction. In: *12th Intl.Conf. On Machine learning*, pp. 115–123 (1995)
- [12] Gaines, B.R., Cronpton, P.: Induction of Ripple-Down rules applied to modelling large databases. *Journal of Intelligent information system* 5(3), 221–228 (1995)
- [13] Panda, M., Patra, M.R.: Ensembling rule based classifiers for detecting network intrusions. In: *International conference on advances in recent techniques communication techniques (ARTCOM 2009)*, Kerla, India. IEEE Computer Society Press, USA (2009)
- [14] Russel, S.J., Norvig, P.: *Artificial Intelligence: A modern approach*. International Edition. Pearson US Imports and PHIPES, London (2002)
- [15] Domingos, P., Pizzani, M.J.: On the optimality of the simple Bayesian classifier under zero-one loss. *Mach.learning* 29(2-3), 103–130 (1997)
- [16] Hall, M., Frank, E.: Combining Naïve Bayes and Decision Tables. In: Wilson, D.L., Chad, H. (eds.) *Proc. of the 21st Intl. Florida Artificial Intelligence society conference (FLAIRS)*, pp. 318–319. AAAI Press, Menlo Park (2008)

Speaker Recognition Using Pole Distribution of Speech Signals Obtained by Bagging CAN2

Shuichi Kurogi, Seitaro Sato, and Kota Ichimaru

Kyushu Institute of technology, Tobata, Kitakyushu, Fukuoka 804-8550, Japan
{kuro@, satou@kurolab.}cntl.kyutech.ac.jp
<http://kurolab.cntl.kyutech.ac.jp/>

Abstract. A method for speaker recognition which uses feature vectors of pole distribution derived from the piecewise linear predictive coefficients obtained by the bagging CAN2 (competitive associative net 2) is presented. The CAN2 is a neural net for learning efficient piecewise linear approximation of nonlinear function, and the bagging CAN2 has been shown to have a stable performance in reproduction and recognition of vowel signals. After training the bagging CAN2 with the speech signal of a speaker, the present method obtains a number of poles of piecewise linear predictive coefficients which are expected to reflect the shape and the scale of the speaker's vocal tract. Then, the pole distribution is used as the feature vector for the speaker recognition. The effectiveness is examined and validated with real speech data.

Keywords: speaker recognition, feature vector of pole distribution, bagging CAN2.

1 Introduction

The competitive associative net called CAN2 has been introduced for learning efficient piecewise linear approximation of nonlinear function [1,2] by means of using the competitive and associative schemes [3,4]. The effectiveness has been shown in several applications; especially, the method using the CAN2 has been awarded the regression winner at the Evaluating Predictive Uncertainty Challenge held at NIPS2004 [5]. In the application to learning and analyzing chaotic and vowel time-series, the time-series is shown to be reproduced with high precision, where multiple piecewise linear predictive coefficients learned by the CAN2 are used for multistep prediction at each consecutive time step [6,7]. Furthermore, the pole distribution derived from the piecewise linear predictive coefficients is shown to be useful for vowel recognition [8] and the bagging version of the CAN2 is shown to have stabler performance [9].

On the other hand, among the conventional researches of speaker recognition including verification and identification, the most common way to characterize the speech signal is short-time spectral analysis, such as Linear Prediction Coding (LPC) and Mel-Frequency Cepstrum Coefficients (MFCC) [10,11,12,13]. Namely, both methods extract multidimensional features from each of consecutive intervals of speech, where a speech interval spans 10-30ms of the speech signal which is called a frame of speech.

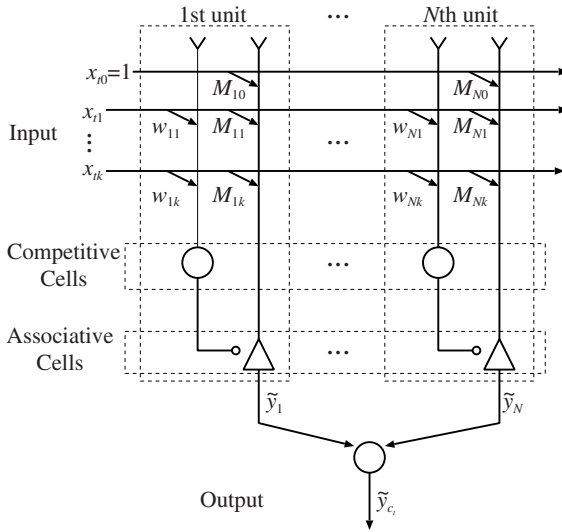


Fig. 1. Schematic diagram of a single CAN2

Thus, a single feature vector of the LPC or the MFCC corresponds to the average of multiple piecewise linear predictive coefficients of the CAN2, which indicates that the CAN2 has stored more precise information on the speech signal.

The remainder is organized as follows; Section 2 gives a brief overview of the single and the bagging CAN2. Section 3 shows the present method for speaker recognition using the pole distribution of predictive coefficients. Section 4 shows the experiments with real speech data and examines the effectiveness.

2 Single and Bagging CAN2

Let $D^n \triangleq \{(\mathbf{x}_t, y_t) \mid t \in I^n\}$ be a given training dataset, where $\mathbf{x}_t \triangleq (x_{t1}, x_{t2}, \dots, x_{tk})^T$ and y_t denote an input vector and the target scalar value, respectively, and $I^n \triangleq \{1, 2, \dots, n\}$ is the index set of the dataset. Here, we suppose the relationship given by

$$y_t \triangleq r_t + \epsilon_t = f(\mathbf{x}_t) + \epsilon_t \tag{1}$$

where $r_t \triangleq f(\mathbf{x}_t)$ is a nonlinear function of \mathbf{x}_t , and ϵ_t represents noise.

A single CAN2 has N units (see Fig. 1). The i th unit has a weight vector $\mathbf{w}_i \triangleq (w_{i1}, \dots, w_{ik})^T \in \mathbb{R}^{k \times 1}$ and an associative matrix (or a row vector) $\mathbf{M}_i \triangleq (M_{i0}, M_{i1}, \dots, M_{ik}) \in \mathbb{R}^{1 \times (k+1)}$ for $i \in I^N \triangleq \{1, 2, \dots, N\}$. The CAN2 approximates the above function $f(\mathbf{x}_t)$ by

$$\hat{y}_t \triangleq \hat{f}(\mathbf{x}_t) \triangleq \tilde{y}_{c_t} \triangleq \mathbf{M}_{c_t} \tilde{\mathbf{x}}_t, \tag{2}$$

where $\tilde{\mathbf{x}}_t \triangleq (1, \mathbf{x}_t^T)^T \in \mathbb{R}^{(k+1) \times 1}$ denotes the (extended) input vector to the CAN2, and $\tilde{y}_{c_t} = \mathbf{M}_{c_t} \tilde{\mathbf{x}}_t$ is the output value of the c_t th unit of the CAN2. The index c_t indicates the selected unit who has the weight vector \mathbf{w}_{c_t} closest to \mathbf{x}_t , or

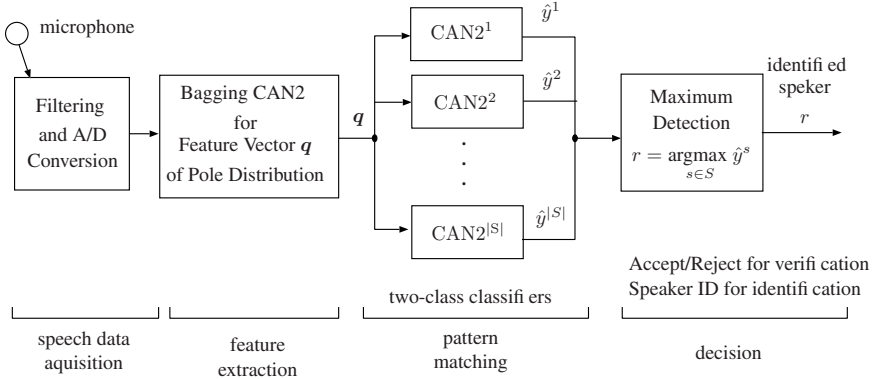


Fig. 2. Speaker recognition system using the CAN2s

$$c_t \triangleq \underset{i \in I^N}{\operatorname{argmin}} \|\mathbf{x}_t - \mathbf{w}_i\|. \tag{3}$$

The above function approximation partitions the input space $V \in \mathbb{R}^k$ into the Voronoi (or Dirichlet) regions $V_i \triangleq \{\mathbf{x} \mid i = \underset{l \in I^N}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{w}_l\|\}$ for $i \in I^N$, and performs piecewise linear approximation of $f(\mathbf{x})$. Note that we have developed an efficient batch learning method (see [2] for details), which we also use in this application.

The bagging (bootstrap aggregation) method [14] is known to have an ability to reduce the variance of the prediction by a single learning machine, and we introduce it into the CAN2: Let $D_l^{\alpha n^*}$ be the l th bootstrap sample set (multiset, or bag) involving αn elements, where the elements in $D_l^{\alpha n^*}$ are resampled randomly with replacement from the given training dataset D^n , where $l \in I^b \triangleq \{1, 2, \dots, b\}$, $\alpha > 0$, and we use $\alpha = 0.7$ and $b = 20$ in the experiments shown below. The bagging prediction of the target value $r_t = f(\mathbf{x}_t)$ is done by the arithmetic mean given by

$$\hat{y}_t^{b^*} \triangleq \frac{1}{b} \sum_{l \in I^b} \hat{y}_t^l = \frac{1}{b} \sum_{l \in I^b} M_{c_t^l} \tilde{\mathbf{x}}_t \tag{4}$$

where $\hat{y}_t^l = M_{c_t^l} \tilde{\mathbf{x}}_t$ is the prediction by the l th CAN2 which has been trained with $D_l^{\alpha n^*}$, and c_t^l is the index of the selected unit in the l th CAN2.

3 Speaker Recognition Using Pole Distribution

3.1 Overview of Speaker Recognition

Fig. 2 shows the present speaker recognition system using the CAN2s. The speaker recognition system, in general, consists of four steps: speech data acquisition, feature extraction, pattern matching, and making a decision. Furthermore, the speaker recognition can be classified into verification and identification, where the former is the process of accepting or rejecting the identity claim of a speaker, which is regarded as two-class

classification. The latter, on the other hand, is the process of determining which registered speaker provides a given utterance, which is regarded as multi-class classification. In addition, speaker recognition has two schemes: text-dependent and text-independent schemes. The former require the speaker to say key words or sentences with the same text for both training and recognition phases, whereas the latter do not rely on a specific text being spoken. In order to deal with both schemes, the present method embeds the features of a text into a feature vector representing the pole distribution derived from the bagging CAN2.

3.2 Model of Speech Signal Production

The most standard model of the speech production is the all-pole linear prediction model described as follows (see [10]); a speech output signal y_t at a discrete time t is modeled by a linear combination of its past values $\mathbf{x}_t = (y_{t-1}, y_{t-2}, \dots, y_{t-k})^T$ as

$$y_t = \mathbf{a}^T \mathbf{x}_t + Gu_t \tag{5}$$

where $\mathbf{a} = (a_1, a_2, \dots, a_k)^T$ represents the predictor coefficients, G is a gain scaling factor, and u_t is the input to the vocal system. In speech application, the input u_t is unknown and k is called prediction order (usually not k but p is used).

On the other hand, the speech signal in the present research is modeled by a more general expression as shown in Eq. (1), and the piecewise linear prediction by the bagging CAN2 given by Eq. (4), where we use $\mathbf{x}_t = (y_{t-1}, y_{t-2}, \dots, y_{t-k})^T$ as for the above standard model. Now, let us rewrite the prediction by the l th CAN2 involved in Eq. (4), or $\hat{y}_t^l = M_{c_t^l} \tilde{\mathbf{x}}_t$, as

$$y_t = M_i^l (1, y_{t-1}, y_{t-2}, \dots, y_{t-k})^T \tag{6}$$

where we replace \hat{y}_t^l by y_t and $M_{c_t^l}$ by M_i^l because we put less importance on the bag number, l , of the prediction \hat{y}_t^l , and the selected unit number, c_t^l , in the following stream. Then, firstly, from [8], we can say that the bagging CAN2 has an ability to store almost all information of vowel signal into \mathbf{w}_i^l and M_i^l for $l \in I^b$ and $i \in I^N$ because the bagging CAN2 after learning a vowel signal could achieve a high-quality reproduction of the vowel by the multistep prediction. Since M_i^l executes the above prediction, M_i^l is supposed to be restricted by the vocal tract and has some information on the vocal tract. Furthermore, if a text speech signal involving many vowels and consonants is trained, we expect that M_i^l ($l \in I^b, i \in I^N$) store the information on the vocal tract of the speaker which is not restricted by a specific vowel or consonant. On the other hand, \mathbf{w}_i^l is supposed to be less restricted by the vocal tract, because it is only used for selecting the unit as shown in Eq. (3).

3.3 Pole Distribution

The values of $M_{i_m}^l$ as well as a_m in Eq. (5) for $m \in I^k \triangleq \{1, 2, \dots, k\}$ are unstable for the change of the prediction order k even if the prediction error is very small, which indicates that they do not express the information of the vocal tract directly and they are

not appropriate for feature vectors in speaker recognition. So, we apply the z -transform to Eq. (6), and we have

$$Y(z) = \frac{M_{i0}^l}{1 - \sum_{m=1}^k M_{im}^l z^{-m}} = \frac{M_{i0}^l}{\prod_{m=1}^k (1 - p_{im}^l/z)} \tag{7}$$

where p_{im}^l are the poles of $Y(z) = Z(y_t)$. Although the poles are expected to reflect the shape and the scale of the vocal tract of the speaker, not the poles directly but the features processed much more, such as the LSP (line spectrum pair) frequencies, MFCC (Mel-frequency cepstrum coefficients), etc. are usually used for speaker recognition. This is mainly because they are demonstrated to work well in speaker-recognition as well as speech-recognition [10]. However, in our empirical results [8], the above poles of the bagging CAN2 are shown to work well in vowel recognition although the poles of only linear coefficients a_m do not work so much.

So, we try to utilize the pole distribution derived from the bagging CAN2 for the feature vector as follows; first, we express the pole by the polar form as $p_{im}^l = r_{im}^l \exp(-j\theta_{im}^l)$, where r_{im}^l is the magnitude, θ_{im}^l is the argument, and $j^2 = -1$. Next, we evenly divide the ranges of the magnitude, $[0, r_{\max}]$, and the argument, $[0, \pi]$, into n_r and n_θ regions, respectively. Then, by counting the number of poles in each region by raster scan from smaller magnitude and smaller argument, we obtain $k_q = n_r \times (n_\theta + 1)$ -dimensional feature vector, e.g. $\mathbf{q} = (q_1, q_2, \dots, q_{k_q})$. Here, note that we set the s th region for the argument as $[(s - 1)\pi/n_\theta, s\pi/n_\theta)$ for $s = 1, 2, \dots, n_\theta$, the number of the poles for $\theta_{im}^l = \pi$ are counted additionally and the obtained n_r elements are augmented to the last part of the feature vector. Furthermore, we neglect the poles with negative imaginary parts, because the pole distribution is symmetric with respect to the real axis on the z -plane. Examples of the pole distributions and feature vectors are shown in Fig. 3. In this figure, we can see that there are oscillatory poles out of the unit circle. Although the vocal tract usually is supposed to be passive, but we have no reason

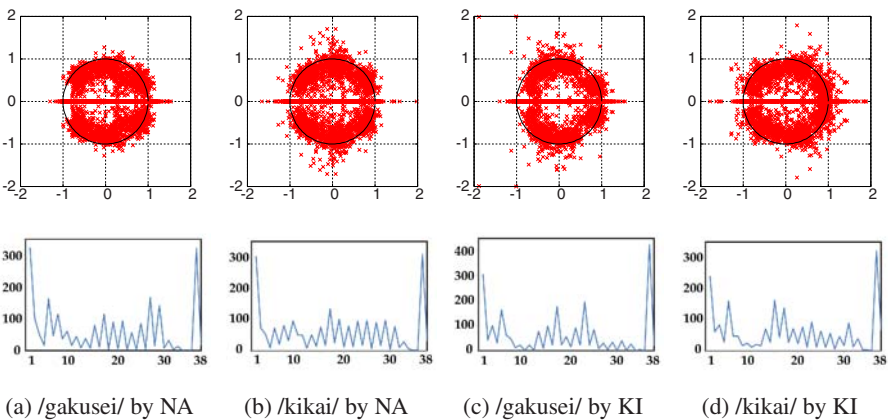


Fig. 3. Examples of pole distributions (upper) and the feature vectors (lower) for the Japanese words /gakusei/ and /kikai/ by speakers NA and KI ($n_r = 2, n_\theta = 18, r_{\max} = 2$)

to neglect those poles. Actually, those poles obtained are necessary for reproducing the high-quality vowel signal via the bagging CAN2 [8], so we dare say that they may be some active or dynamic characteristic of the vocal tract. Furthermore, note that the fluctuation of the feature vector in Fig. 3 does not indicate the pitch component of the speech as usually appeared in the vowel spectrum, but the effect of the raster scan from smaller to bigger magnitude.

3.4 Pattern Matching for Speaker Verification and Identification

For the pattern matching and the classification, we utilize multiple CAN2s, where each CAN2 is used for verifying a speaker. Here, note that the CAN2 basically is for regression problems but we can use it as a two-class classifier by means of binarizing the output of the CAN2. Furthermore, although the bagging CAN2 is also applicable, we explain the case of the single CAN2 in the following: First, let S and U be the sets of speakers and texts, respectively, Q^s be the set of feature vectors \mathbf{q} of the pole distribution for the speaker s . Then, for CAN2^s, or the CAN2 as a two-class classifier for the speaker s , we replace the system equation shown in Eq. (1) by $y^s = f^s(\mathbf{q}) + \epsilon_{\mathbf{q}}$, where $\epsilon_{\mathbf{q}}$ indicates the effect of the variation of \mathbf{q} , and

$$f^s(\mathbf{q}) = \begin{cases} 1, & \text{if } \mathbf{q} \in Q^s, \\ -1, & \text{otherwise.} \end{cases} \quad (8)$$

Moreover, for the error calculation of the learning method [2], we binarize the output of the original CAN2 given by Eq. (2) as

$$\hat{z}^s = \begin{cases} 1, & \text{if } \hat{y}^s = M_c^s \tilde{\mathbf{q}} \geq 0, \\ -1, & \text{otherwise.} \end{cases} \quad (9)$$

where M_c^s is the associative memory of the selected unit c in CAN2^s, and $\tilde{\mathbf{q}} = (1, \mathbf{q}^T)^T$. After training CAN2^s with a certain number of training data $(\mathbf{q}, f^s(\mathbf{q}))$ for $\mathbf{q} \in Q^s$, we can execute a speaker verification for the speaker s by Eq. (9).

Furthermore, we use the original output, $\hat{y}^s = M_c^s \tilde{\mathbf{q}}$, as a score of the pattern matching for the speaker identification. Namely, for an unknown feature vector \mathbf{q} , the speaker identification number r can be obtained by the maximum detection as $r = \operatorname{argmax}_{s \in S} \hat{y}^s$, where CAN2^s for all $s \in S$ are supposed to have been trained.

4 Experimental Results and Discussion

We use the speech signals sampled with 8kHz of sampling rate and 16 bits of resolution in a silent room of our laboratory. They are from five speakers: $S = \{\text{NA, KI, KH, MO, RM}\}$ where NA is female and the others are male. We mainly examined five texts (or Japanese words): /gakusei/, /kikai/, /daigaku/, /kyukodai/, /fukuokaken/, while we also used five Japanese vowels (/a/, /i/, /u/, /e/, /o/) for an analysis shown below. For each speaker and each text, we use ten sets of speech data.

First, we examined the performance in text-dependent speaker recognition. For speaker identification, we apply the leave one set out cross-validation (LOOCV) for

Table 1. Error rate [%] obtained by the LOOCV in various cases of speaker recognition. The variables E_v and E_i are for verification and identification, respectively. For all cases, default parameter values are $k = 8$, $N = 20$ and $b = 20$ for the bagging CAN2, and $n_r = 2$, $n_\theta = 18$ for the feature vectors of pole distribution. (a) is the result of text-dependent recognition for the five texts. (b) is of text-independent recognition using the five texts shown in (a). (c) is obtained by the single CAN2 with $N = 20$. (d) is of text-independent recognition for five vowels. (e) is of the text-independent recognition for /gakusei/ with different n_θ .

(a)			(b)			(d)			(e)		
text	E_v	E_i	text	E_v	E_i	text	E_v	E_i	n_θ	E_v	E_i
/gakusei/	5.6	4.0	independent	5.6	2.0	/a/	4.0	0.0	6	9.6	14.0
/kikai/	6.8	6.0				/i/	12.0	0.0	12	8.4	10.0
/daigaku/	2.4	0.0				/u/	16.0	60.0	18	5.6	4.0
/kyukodai/	1.6	0.0	(c)			/e/	0.0	0.0	24	5.2	8.0
/fukuokaken/	0.4	2.0	text	E_v	E_i	/o/	24.0	100.0	30	5.2	4.0
total	4.1	2.4	/gakusei/	15.2	32.0	total	11.2	32.0	48	5.2	8.0
			(single CAN2)								

the ten sets of each text for five speakers. Namely, we leave one set out for the test, and use the remaining sets for training, and this process is applied to every set for the test. So, there are 50 (=5 speakers \times 10 sets) test data for identification. Through these speaker identification tests, the performance of the speaker validation by each CAN2^s can be obtained. Namely, for every test of 50 data, five CAN2^s for all speakers output the decision, thus we have 250 data for the speaker verification. The result is shown in Table 1(a). We can see that the error rate for the speaker identification, E_i , is zero for two texts, /daigaku/ and /kyukodai/, and a little bit worse result for other words. The error rate for the speaker verification by CAN2^s, E_v , was a little bit worse than E_i .

Next, we examined the performance in text-independent speaker recognition, where we used all of the five texts shown in (a) and two datasets for each speaker and each text. The result is shown in Table 1(b). We can see that the error rates are almost the same as those in (a). This result suggests that the present system could extract the characteristics of the speakers independent from the texts. Moreover, the error rates seem to be competitive with the results by the previous works shown in [10] for the text-independent speaker recognition.

In order to compare with the performance of the single CAN2, we executed the text-dependent recognition for /gakusei/ with the single CAN2. The result is shown in Table 1(c), and we can see that the error rates are much bigger than those for /gakusei/ shown in (a). This result indicates that the lots of poles obtained by the bagging method, as shown in Fig. 3, is supposed to provide a good variation of the poles. This seems to correspond to the fact that the bagging method makes a good variation of predictions and their mean provides a stable and high-quality prediction [14], where only the mean usually is used for prediction but the variation is utilized in this application.

Since the shape of the vocal tract is stable for vowels and it may play an important role in speaker recognition, we executed the text-independent speaker recognition for five vowels with one dataset for each vowel and each speaker. The result is shown in (d), and it suggests that the performance depends on each vowel, but the variance

through the vowels is very large. By comparing with the results in (a), we could not find any single vowel which contributes to $E_i = 0$ or which derives bigger E_i .

We examined the effect of n_θ , or the resolution of the pole distribution for the feature vector by text-dependent recognition for /gakusei/, as shown in Table II(e). From this result, we can see that low resolution $n_\theta < 18$ does not work so well, while high resolution up to $n_\theta = 48$ provides a certain level of the performance.

5 Conclusion

We have presented a method for speaker recognition which uses feature vectors of pole distribution derived from the bagging CAN2. The effectiveness of the method is examined and validated with real speech data. Since the poles correspond to frequency response modes of the vocal tract, they are expected to reflect the shape and the scale of the vocal tract much more than the spectrum and the cepstrum, we would like to compare with such methods theoretically from this point of view in our future research study. Furthermore, since the size of the dataset we examined is small, we would like to use much bigger dataset in our future research.

Acknowledgments. This work was partially supported by the Grant-in Aid for Scientific Research (C) 21500217 of the Japanese Ministry of Education, Science, Sports and Culture.

References

1. Kurogi, S., Ren, S.: Competitive associative network for function approximation and control of plants. In: Proc. NOLTA 1997, pp. 775–778 (1997)
2. Kurogi, S., Ueno, T., Sawa, M.: A batch learning method for competitive associative net and its application to function approximation. In: Proc. SCI 2004, vol. V, pp. 24–28 (2004)
3. Kohonen, T.: Associative Memory. Springer, Heidelberg (1977)
4. Ahalt, A.C., Krishnamurthy, A.K., Chen, P., Melton, D.E.: Competitive learning algorithms for vector quantization. *Neural Networks* 3, 277–290 (1990)
5. <http://predict.kyb.tuebingen.mpg.de/pages/home.php>
6. Kurogi, S., Ueno, T., Tanaka, K.: Asymptotic optimality of competitive associative net and its application to chaotic time series prediction. In: Proc. JNNS 2002, pp. 283–284 (2002)
7. Kurogi, S., Sawa, M.: Analysis of vowel time series via competitive associative nets. In: Proc. JNNS 2003, pp. 54–55 (2003)
8. Kurogi, S., Nedachi, N.: Reproduction and recognition of vowels using piecewise linear predictive coefficients obtained by competitive associative nets. In: Proc. SICE-ICCAS 2006, CD-ROM (2006)
9. Kurogi, S., Nedachi, N., Funatsu, Y.: Reproduction and recognition of vowel signals using single and bagging competitive associative nets. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) ICONIP 2007, Part II. LNCS, vol. 4985, pp. 40–49. Springer, Heidelberg (2007)
10. Campbell, J.P.: Speaker Recognition: A Tutorial. *Proc. the IEEE* 85(9), 1437–1462 (1997)
11. Furui, S.: Speaker Recognition. In: Cole, R., Mariani, J., et al. (eds.) *Survey of the state of the art in human language technology*, pp. 36–42. Cambridge University Press, Cambridge (1998)
12. Hasan, M.R., Jamil, M., Rabbani, M.G., Rahman, M.S.: Speaker identification using Mel frequency cepstral coefficients. In: Proc. ICECE 2004, pp. 565–568 (2004)
13. Bocklet, T., Shriberg, E.: Speaker recognition using syllable-based constraints for cepstral frame selection. In: Proc. ICASSP (2009)
14. Breiman, L.: Bagging predictors. *Machine Learning* 26, 123–140 (1996)

Fast Intra Mode Decision for H.264/AVC Based on Directional Information of I4MB

Kyung-Hee Lee¹, En-Jong Cha², and Jae-Won Suh¹

¹ Chungbuk National University, College of Electrical and Computer Engineering,
12 Gaeshin-dong, Heungduk-gu, Chongju, Korea

khlee82@cbnu.ac.kr, sjwon@cbnu.ac.kr

² Chungbuk National University, Department of Biomedical Engineering,
12 Gaeshin-dong, Heungduk-gu, Chongju, Korea

ejcha@cbnu.ac.kr

Abstract. H.264/AVC video encoder adapting a rate-distortion optimization technique to select the coding mode for each macroblock (MB) gets a higher coding efficiency than those of previous video coding standards but the computational complexity increases drastically. To reduce the computational complexity, we propose a fast intra mode decision algorithm based on directional information of I4MB. Simulation results demonstrate that the proposed algorithm generates generally good performances in PSNR, bit rates, and processing time.

Keywords: H.264/AVC, Fast Intra Mode Decision, Directional Information.

1 Introduction

The fast growth of entertainment services has generated a great deal of interest in the transmission of digitized video data. To transmit vast video data over a band-limited channel, an efficient video coding standard is necessary, such as H.264/AVC [1]. By adopting new coding techniques, the H.264/AVC can generate the high coding efficiency but real time encoding is very difficult due to high computational complexity. To reduce the complexity of H.264/AVC, many fast mode decision methods have been suggested: Fast variable block size motion estimation (ME) [2], fast coding mode selection [3], fast intra prediction [4], etc. Among them, a fast intra mode decision algorithm is a good approach because the computational cost of the intra prediction is comparable to that of the inter prediction.

In this paper, we propose a fast intra mode decision algorithm. It generates a good performance without a large loss of PSNR and a big increment of bit rate. In section 2, we briefly introduce the general mode decision algorithm to help understanding our proposed scheme. Section 3 explains a new fast intra mode decision algorithm based on directional information of I4MB. Finally, we demonstrate the performances of the proposed algorithm and draw conclusions in section 4 and section 5.

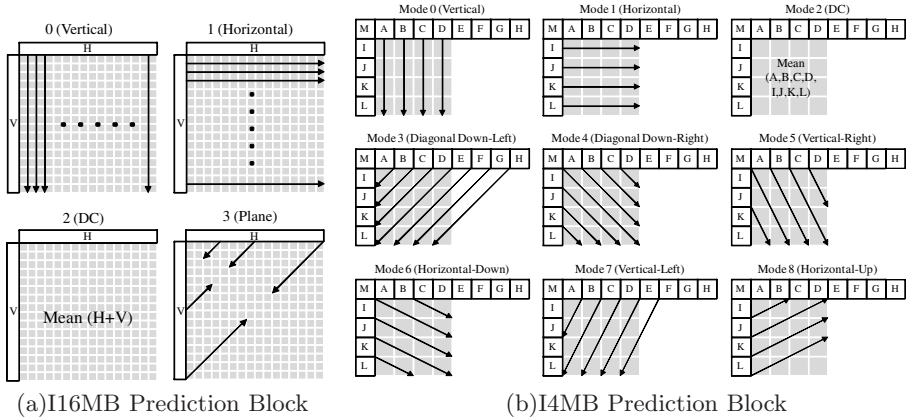


Fig. 1. Intra Prediction Modes

2 MB Prediction Mode for Inter Frame

H.264/AVC adapts the rate-distortion optimization (RDO) technique to determine the best MB coding mode, in terms of minimizing bit rate and maximizing image quality.

2.1 Inter Prediction Mode

Unlike the inter MB encoding mode of previous video coding standards, the MB can be motion estimated and compensated by multi reference frames with varying block sizes from 16×16 down to 4×4 . The luminance component of each MB may be partitioned in four ways and motion compensated either as one 16×16 partition, two 8×16 partitions, or four 8×8 partitions. If the 8×8 partition is chosen, each of the four 8×8 sub-MBs should be further divided into partitions with block sizes of 8×4 , 4×8 , or 4×4 , which are called the SUB 8×8 mode. In inter MB prediction stage, the H.264/AVC predicts the motion vector (MV) and reference frame for each partition.

2.2 Intra Prediction Mode

In addition to the inter MB coding modes, various intra predictive coding modes are specified in H.264/AVC. In contrast to previous video coding standards, intra MB for H.264/AVC is predicted in the spatial domain and residual data are encoded. For the luminance samples, the prediction block can be a 16×16 block (I16MB) or a 4×4 block (I4MB). As shown in Fig. 1, I16MB and I4MB have 4 prediction modes and 9 prediction modes, respectively.

2.3 Best Coding Mode Selection by RDO

To obtain the best MB coding mode for inter frame, H.264/AVC encoder exhaustively tests all possible encoding modes for each MB.

Inter Mode

1. Calculate $mcost$ for 16×16 , 16×8 , and 8×16 blocks

$$mcost = SAD_{mode} + \lambda_{motion} \cdot R(MV, REF), \quad (1)$$

where SAD_{mode} is the sum of absolute differences between the current block and its motion estimated block. λ_{motion} denotes the lagrangian multiplier. $R(MV, REF)$ means bitrates for encoding the MV and notifying the number of reference frame to be used.

2. Calculate RD costs for 8×8 , 8×4 , 4×8 , and 4×4 blocks.

$$J_{RD} = SSD_{mode} + \lambda_{mode} \cdot R(s, r, M), \quad (2)$$

where SSD_{mode} means the squared sum of differences between the current block and its motion compensated block. λ_{mode} is the lagrangian multiplier. $R(s, r, M)$ is bitrates to encode selected mode M , where s is the current block and r is the predicted block.

3. Skip mode check
4. If the current MB is not a SKIP mode, we calculate RD costs for 16×16 , 16×8 , 8×16 blocks
5. Determine the best inter MB mode.

Intra Mode

1. Calculate SAD for 4 prediction modes for I16MB
2. Calculate RD cost for I16MB mode having the minimum SAD
3. Calculate RD costs for all prediction modes for I4MB
4. Determine the best intra MB mode.

Select the best MB coding mode by comparison of RD costs

3 Proposed Fast Intra Mode Decision Algorithm

To propose a fast intra mode decision algorithm, we analyze the spatial correlation within a current block and find that the pixels along the local edge direction generally have similar values. If we determine the representative direction of I4MB, a suboptimal RD cost can be obtained around the direction. Therefore, we can skip several prediction modes. In addition, the statistical data of RD costs of sixteen I4MB can be used to choose the best prediction mode for I16MB, because I16MB is composed of sixteen I4MB blocks. Consequently, we can save the processing time by reduction of the number of RD calculation.

3.1 Mode Prediction for I4MB

First, SAD values for main four direction modes are calculated as shown in Fig. 3, which are expressed in (3). The mode having the smallest SAD is determined as

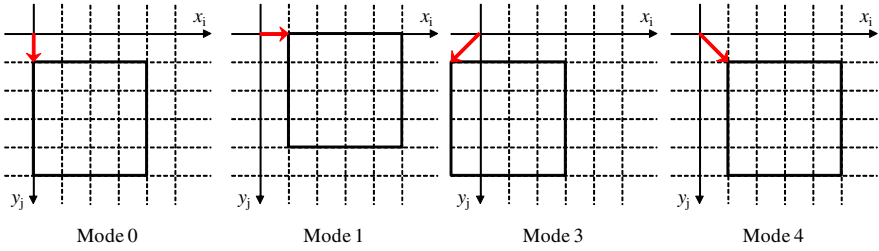


Fig. 2. Directional Mask for I4MB Prediction

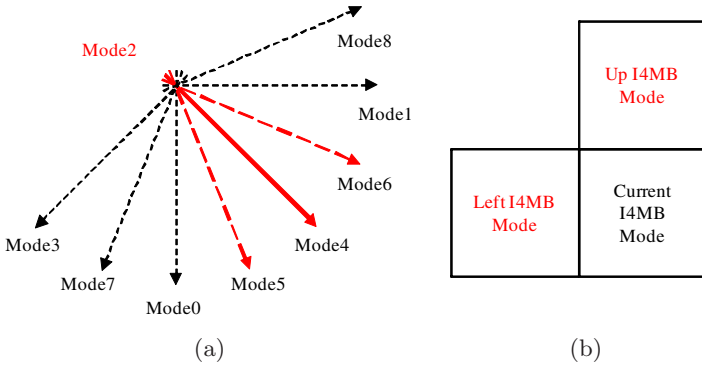


Fig. 3. Candidates for I4MB Prediction

a primary direction mode (PDM) of the current I4MB. Mode0, Mode1, Mode3, and Mode4 notify vertical(\downarrow), horizontal(\rightarrow), backward diagonal(\swarrow), and forward diagonal(\searrow) directions, respectively. Due to the directional property, the smallest RD cost can be generated around the PDM.

$$\begin{aligned}
 Mode0 \quad SAD_0 &= \sum_{j=0}^3 \sum_{i=0}^3 |x_i y_j - x_i y_{j+1}| \\
 Mode1 \quad SAD_1 &= \sum_{j=0}^3 \sum_{i=0}^3 |x_i y_j - x_{i+1} y_j| \\
 Mode2 \quad SAD_2 &= \sum_{j=0}^3 \sum_{i=0}^3 |x_i y_j - x_{i-1} y_{j+1}| \\
 Mode3 \quad SAD_3 &= \sum_{j=0}^3 \sum_{i=0}^3 |x_i y_j - x_{i+1} y_{j+1}| \tag{3}
 \end{aligned}$$

Second, RD costs are calculated. To select more precise prediction mode for I4MB, we add three auxiliary modes for calculating RD cost, such as Mode2 (DC) and

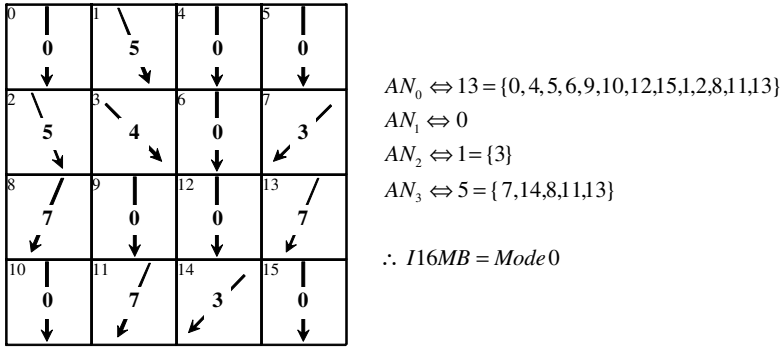


Fig. 4. Mode decision for I16MB

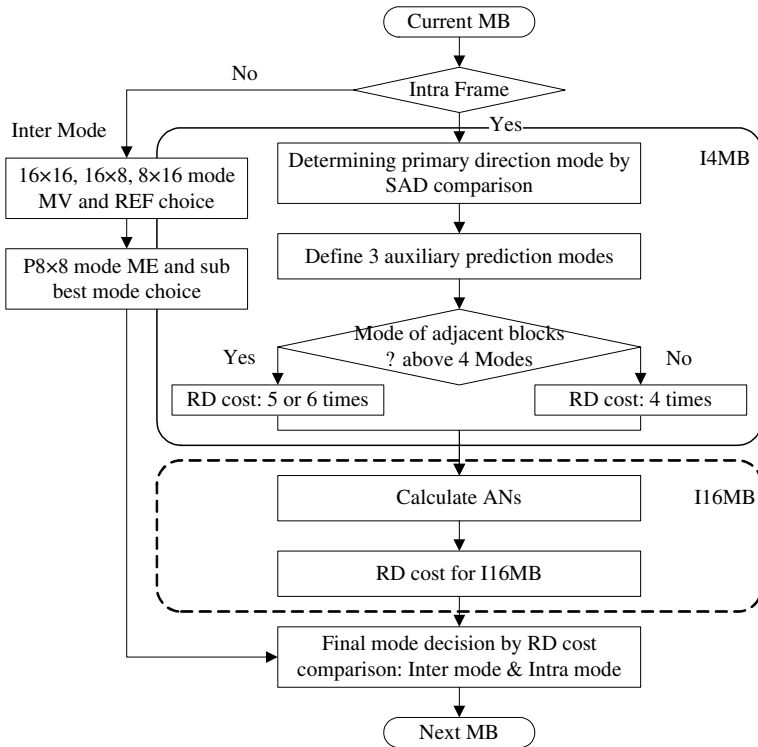


Fig. 5. Flowchart of the proposed algorithm

both side modes of PDM as shown in Fig. 3(a). In addition, we consider the spatial correlation of mode information. As shown in Fig. 3(b), if up and left I4MB are already determined as one of the previously determined prediction modes, we do not need to calculate any other RD costs. However, if only one or none of them is included in those modes, we calculate RD costs for the adjacent prediction modes.

3.2 Mode Prediction for I16MB

Because I16MB is composed of sixteen I4MB blocks, the statistical data of RD costs of sixteen I4MB can be used to choose the best prediction mode for I16MB. The accumulated number (AN) of each mode for I4MB is compared, and then the mode having the maximum value is determined as a best candidate mode for I16MB. Because the number of the prediction mode for I16MB is smaller than that of I4MB, there is a rule to separation, which is expressed in (7). Fig. 4 shows an example for defining the mode for I16MB.

Finally, according to which one generates the smaller RD cost, I4MB or I16MB is determined as an intra coding mode for current MB. To implement the proposed algorithm, we modify the H.264/AVC encoder structure. The flowchart for the proposed intra mode decision algorithm is shown in Fig. 5.

4 Simulation Results

Our propose methods have been implemented and compared with that of the JVT reference software JM11.0. Simulation conditions are summarized into Table 1.

Our proposed algorithm was compared with Pan et al. algorithm [4] under the same conditions. We used several measures for evaluating the performance of the proposed algorithm.

$$\Delta PSNR = PSNR_{new_method} - PSNR_{JM} \quad (4)$$

Table 1. Simulation Conditions

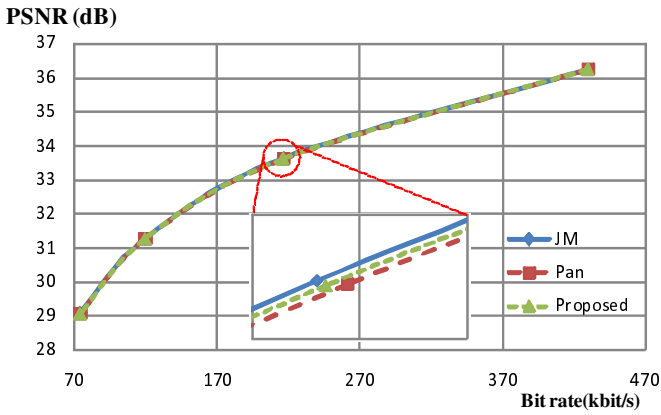
Parameters	Condition
Profile	Main
Search Range	32
Number of Reference Frames	1
Sequence Types	IPPP and All I frames
Interval between I frames	100

Table 2. Performance Comparison for IPPP Sequences

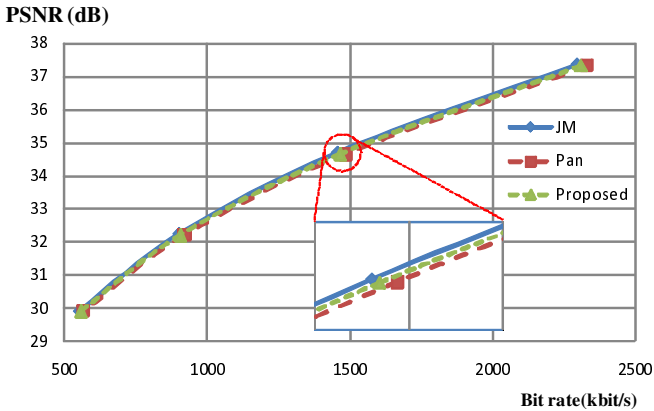
Sequence		QP=28			QP=32		
		ΔP	ΔB	ΔT	ΔP	ΔB	ΔT
Akiyo	Pan et al. [4]	0.00	0.42	-13.39	0.00	0.82	-11.99
	Proposed	0.00	-0.17	-13.62	-0.01	0.58	-12.40
City	Pan et al. [4]	0.00	0.18	-14.98	0.00	1.02	-13.46
	Proposed	0.00	0.00	-13.70	0.00	0.47	-12.17
Foreman	Pan et al. [4]	0.00	0.22	-15.98	-0.02	0.19	-16.79
	Proposed	-0.01	0.05	-16.72	-0.02	-0.16	-16.55
Stefan	Pan et al. [4]	-0.01	0.23	-17.17	0.00	0.37	-16.30
	Proposed	-0.01	0.08	-15.89	0.00	0.07	-14.33

Table 3. Performance Comparison for All I Frames

Sequence		QP=28			QP=32		
		ΔP	ΔB	ΔT	ΔP	ΔB	ΔT
Akiyo	Pan et al. [4]	0.00	0.42	-13.39	0.00	0.82	-11.99
	Proposed	0.00	-0.17	-13.62	-0.01	0.58	-12.40
City	Pan et al. [4]	0.00	0.18	-14.98	0.00	1.02	-13.46
	Proposed	0.00	0.00	-13.70	0.00	0.47	-12.17
Foreman	Pan et al. [4]	0.00	0.22	-15.98	-0.02	0.19	-16.79
	Proposed	-0.01	0.05	-16.72	-0.02	-0.16	-16.55
Stefan	Pan et al. [4]	-0.01	0.23	-17.17	0.00	0.37	-16.30
	Proposed	-0.01	0.08	-15.89	0.00	0.07	-14.33



(a) IPPP sequence



(b) All intra frames

Fig. 6. Comparisons of Rate Distortion Curves

$$\Delta Bits = \frac{Bit_{new_method} - Bit_{JM}}{Bit_{JM}} \times 100(\%) \quad (5)$$

$$\Delta Time = \frac{Time_{new_method} - Time_{JM}}{Time_{JM}} \times 100(\%) \quad (6)$$

We summarized the performance of fast intra mode decision algorithms in Table 2 and Table 3. The proposed algorithm achieves consistent time savings about 15% in IPPP sequence and 44% in all I frames with negligible PSNR loss and small increment of bit rate compared with JM11.0. Especially, interesting observation of all I sequences is that the proposed algorithm reduces the increment of bit rate compared with Pan's method. As shown in Fig. 6, the RD results of proposed algorithm generate the similar to those of JM11.0.

$$\begin{aligned} AN_0 &= \text{the number of } Mode0, Mode5, Mode7 \\ AN_1 &= \text{the number of } Mode1, Mode6, Mode8 \\ AN_2 &= \text{the number of } Mode2, Mode4 \\ AN_3 &= \text{the number of } Mode3, Mode7, Mode8 \end{aligned} \quad (7)$$

5 Conclusions

This paper presented a fast intra mode decision for H.264/AVC encoder by reducing the number of RDO calculation based on directional information of I4MB. By using statistical data of RD costs for sixteen I4MB, we can easily decide the best prediction mode for I16MB. The proposed algorithm reduces the processing time about 15% in IPPP sequences and 44% in all I frame sequences with negligible loss in PSNR and small increment of bit rate compared with those of JM11.0.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2009-0063258).

References

1. Information Technology-Coding of Audio-Visual Objects-part 10: Advanced Video Coding, Final Draft International standard, ISO/IEC FDIS 14496-10 (March 2005)
2. Kuo, T.Y., Chan, C.H.: Fast Variable Block Size Motion Estimation for H.264 Using Lilelihood and Correlation of Motion Field. IEEE Trans. Circuit and Systems for Video Technology 16, 1185–1195 (2006)
3. Choi, I., Lee, J., Jeon, B.: Fast coding mode selection with rate-distortion optimization for MPEG-4 part-10 AVC/H.264. IEEE Trans. Circuit and Systems for Video Technology 16(12), 1557–1561 (2006)

4. Pan, F., Lin, X., Rahardja, S., Lim, K.P., Li, Z.G., Wu, D., Wu, S.: Fast Mode Decision Algorithm for Intraprediction in H.264/AVC Video Coding. IEEE Trans. Circuit and Systems for Video Technology 15(7), 813–822 (2005)
5. http://iphone.hhi.de/suehring/tml/download/old_jm/jm11.0.zip

Palmprint Recognition Based on Local DCT Feature Extraction

H. Kipsang Choge, Tadahiro Oyama, Stephen Karungaru,
Satoru Tsuge, and Minoru Fukumi

The University of Tokushima, 2-1 Minami-Josanjima,
Tokushima 770 - 8506, Japan
choge@is.tokushima-u.ac.jp

Abstract. In this paper we present a method which extracts features from palmprint images by applying the Discrete Cosine Transform (DCT) on small blocks of the segmented region of interest consisting of the middle palm area. The region is extracted after careful preprocessing to normalize for position and illumination. This method takes advantage of the well known capability of the DCT to represent natural images using only a few coefficients by performing the DCT on each block. After ranking the coefficients by magnitude and selecting only the most prominent, these are then concatenated into a compact feature vector that represents each palmprint. Recognition and verification experiments using the PolyU Palmprint Database show that this is an effective and efficient approach, with a recognition rate above 99 % and Equal Error Rate (EER) of less than 3 %.

Keywords: palmprint authentication, Discrete Cosine Transform, block DCT, local feature extraction, hand-based biometrics.

1 Introduction

Biometrics offer an attractive alternative to traditional token-based methods of personal identification or verification because they are harder to circumvent and easier to use [1].

The print patterns on the palm are not duplicated in other people, even in mono zygotic or identical twins, and they do not change over the entire lifetime of an individual. The palm is also larger in area than the iris and fingerprint, and contains an abundance of features which include principal lines, wrinkles, creases and minutiae points. This makes it possible to use lower resolution images for palmprint recognition, increasing speed and lowering cost. Also, palmprint biometrics have a higher user acceptance than those which use iris scans because the palmprint image capture process is not as invasive [2,3].

This paper describes a novel method of palmprint feature extraction and matching using block-based Discrete Cosine Transform (DCT) that, combined with careful preprocessing and choice of block size, has proved to be quite effective in palmprint discrimination.

Research in palmprint recognition can be divided into two main categories: statistical and structural methods [2]. Statistical methods transform the palmprint into a new space or consider it as a point in a multidimensional space, while structural methods involve the extraction of information from structural features of palmprints such as principal lines, ridges and creases. Statistical methods reported in the literature include Eigenpalms [14], Fisherpalms [5], those based on local and global texture [3], using Gabor filters [6], and Fourier Transform [17]. Structural methods include those that extract principal lines [8], extraction of features based on palm creases [9], extraction of structural features using wavelets [10], and the use of hand geometry features [11].

The Karhunen-Loeve transform (KLT) is used in [1] and [4] to produce an expansion of the input image in terms of a set of basis images or “Eigenpalms”. Using only a portion of the KLT coefficients, impressive recognition results are achieved. Fishers Linear Discriminant Analysis is applied on a set of palmprints to find the optimal linear transformation that maximizes the Fisher criterion in [5]. The KLT transform is first applied to guarantee non-singularity and the method performs as well as Eigenpalms, although both are computationally expensive and data-dependent.

In [1] and [7], the DFT image of a palmprint is divided into ring-like portions centered at the zero frequency point to extract features representing frequency, and into slices cutting through the middle to extract features representing direction. Both are then used for recognition with an accuracy of about 96%.

Hafed and Levine used a subset of the global DCT coefficients of an input face, and by using an affine transformation to correct for scale and orientation together with illumination normalization proved that their method could perform as well as the benchmark Eigenfaces technique in recognition and verification [13]. In [14], block-based DCT similar to that used in the JPEG standard [15] is applied to achieve good face recognition results using less than half the DCT coefficients.

In a face and palmprint fusion method, [16] used global DCT coefficients and selected specific frequency bands based on a 2-D separability judgment. A frequency band is judged to be an effective means of class separation by evaluating the ratio of the between-class scatter to the within-class scatter for the entire data set when the specific band is used as a feature vector. If the ratio is 1 or more, the frequency band is judged to have good linear separability. The approach is similar to performing Fisherpalms [5] using just the chosen frequency bands as input instead of the individual palms, making the method rather complicated.

1.1 Proposed Approach and Motivation

The main merit of the DCT is its close relationship to both the KLT and DFT. In an information packing sense, the KLT is the optimal transform because it minimizes the mean square error, but the DCT follows very closely behind it [12][13]. Unlike the KLT, however, the DCT is data-independent and can be evaluated using the same fast algorithms used for obtaining the DFT. It also has

the major advantage of having been implemented in a single integrated circuit unlike other data-independent transforms like the DFT [12].

These factors motivated us to explore the use of DCT coefficients for palmprint identification after our earlier work showed that careful selection of a subset of DFT coefficients combined with proper preprocessing can produce very good palmprint recognition results [17]. We propose a method that uses only the largest 50 % or less of the DCT coefficients from non-overlapping blocks of the input image. This ensures that local information about the features in the palmprint image is retained. The blocks are either 8×8 or 16×16 pixels in size and classification is based on the simple Euclidean distance. Careful preprocessing is performed on the images prior to extracting the central palm area in order to improve performance. We compare our results to other transform-based methods in the literature that used images from the same database and demonstrate the effectiveness of our method.

2 Palmprint Recognition Using Local DCT Features

An overview of the proposed system is shown in Fig. 1. The input consists of uncompressed gray scale bitmap images of the palm scanned at a resolution of 75 dots per inch and with an original size of 384×284 pixels. Before extracting the square central palm area, it is necessary first to obtain a reference coordinate system to ensure that approximately the same area is extracted from each image. This is done together with brightness normalization in the preprocessing stage as shown in Fig. 2. A square 128×128 pixel area is then extracted from the middle palm area and this represents the input image fed into the block-based feature extraction stage of the system.

In the block based feature extraction stage, two major experiments are carried out. First, local DCT coefficients are calculated from the original 128×128 input image using 8×8 or 16×16 pixel blocks and after ranking based on magnitude, the top half are selected and concatenated from each block to form the feature vector for each input image. The aim of using the two block sizes is to determine the optimal block size for performing DCT when the input image size is fixed at 128×128 pixels. Because the JPEG standard [15] uses an 8×8 block size, using the same size would be ideal for integration with a system that uses JPEG-compressed palmprint images, but we wish to compare the results with those obtained using a 16×16 block size as well.

The second block DCT feature extraction experiment involves reducing the original 128×128 input image via bi-cubic interpolation to 64×64 and 32×32 pixels and extracting DCT features using a fixed block size in order to determine the effect of input image size reduction on recognition accuracy for each of the two block sizes. The effect of gradual size reduction has not been investigated rigorously enough in other transform-based methods in the literature, even though a mere 50 % reduction in input image size would increase the speed of computation for such a system by a factor of 4. The results obtained from the two experiments can therefore be evaluated based on optimizing not only the

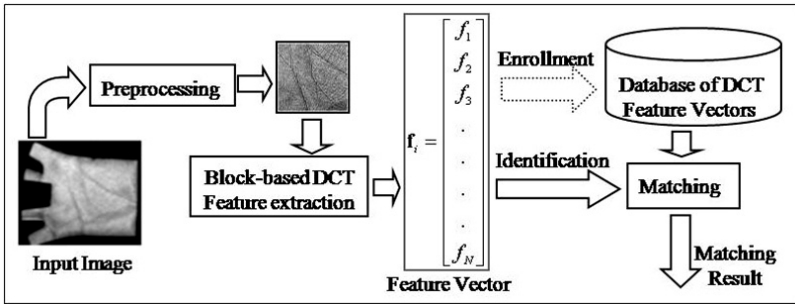


Fig. 1. System architecture for palmprint recognition via block-based DCT

block size used for DCT feature extraction and the input image size but also on the number of coefficients retained per block when forming the feature vectors. In this manner, the final configuration selected will be the one that involves using the most compact feature vector to produce the best recognition results.

In the matching stage, a classifier is used based on minimizing the distance D , which is the summation of the Euclidean distance between the feature vector from each block of the unknown palm b and that from the corresponding block in the database image a

$$D = \sum_{i=1}^N \sqrt{\sum_{j=1}^L (a_{ij} - b_{ij})^2} . \tag{1}$$

where N is the number of blocks per image and L is the length of the feature vector per block and corresponds to the number of coefficients retained.

2.1 Preprocessing

The method proposed and explained in great detail by [11] is used to normalize the input image for position and rotation before extracting the central palm area. The two points k_1 and k_2 shown in Fig. 2 (a) are located on the image and used as the reference x axis. A perpendicular line originating from the midpoint m between these two points is then used as the y axis. The square region shown starts at a fixed distance from point m . Because the points k_1 and k_2 are determined based on the center of the gravity of the corresponding hole between fingers, this region is approximately the same in each image.

Brightness normalization is performed on the extracted region-of-interest (ROI), summarized in Fig. 2 (b)–(e). A coarse estimate of the background illumination is first obtained by finding the average of every 8×8 region over the entire ROI [18]. Each average value is a pixel in a 16×16 image that is expanded to the original ROI size via bi-cubic interpolation. The estimated background is then subtracted from the original image followed by contrast improvement by local histogram equalization using 32×32 blocks as shown in Fig. 2(e).

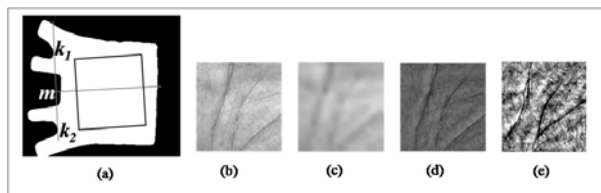


Fig. 2. (a) The coordinate system used for position normalization and the central palm area to be extracted, (b) extracted 128×128 palm image, (c) estimated background image, (d) background-subtracted image, and (e) after local histogram equalization

2.2 Block DCT Feature Extraction

Feature vectors that represent the palms in the database are formed by computing the block-based DCT of the processed images. The 2D DCT, $C(u, v)$, of an image $I(r, c)$ whose size is $N \times N$ pixels is given by

$$C(u, v) = \alpha(u)\alpha(v) \sum_{r=0}^{N-1} \sum_{c=0}^{N-1} I(r, c) \cos \left[\frac{(2r + 1)u\pi}{2N} \right] \cos \left[\frac{(2c + 1)v\pi}{2N} \right] , \quad (2)$$

where $\alpha(u)$ and $\alpha(v)$ are given by

$$\alpha(u), \alpha(v) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u, v = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u, v = 1, 2, \dots, N - 1 \end{cases} . \quad (3)$$

Using equation 2, a vector is formed for each 8×8 or 16×16 block by extracting the DCT coefficients, ranking from the smallest to the largest based on the magnitude, and retaining only the first 50%. The DC value represents the mean of the pixel values within a given block and is included in the feature vector as well. Each image is therefore represented in feature space by a matrix whose columns are the sorted magnitude values and whose dimensions are $L \times N$, given in equation 1.

3 Experiments, Results and Analyses

Palmprint images from the Hong Kong Polytechnic University palmprint database ('PolyU Palmprint Database') [19] were used in our experiments. In this database, between 7 and 10 images of the left and right palms are captured in bitmap format per individual in each of two sessions, approximately 2 months apart.

For recognition, 2 images each for 33 palms from the first session are selected at random. A training template is formed by averaging the feature vectors from each corresponding block of the processed images. The test set consists of 4 images for each of the same palms from the second session for a total of 132 images belonging to 33 classes. The test set is used for recognition by matching

each of the 132 feature vectors to every template in the training set, and based on the total Euclidean distance, a test sample belongs to the same class as the template with which the minimum distance is obtained. A total of 33 matching attempts are performed for each test sample to give a total of 4356 attempts for the whole test set.

For verification, an evaluation set containing 5 randomly selected images each for 33 palms from either session is formed. To test the performance of the system, a one-to-all matching using equation (1) is done for each of the images in the set while varying the decision threshold so that at each threshold, a total of ${}_{165}C_2$ ($165 \times 164 \div 2 = 13530$) matching attempts are made, of which ${}_5C_2 \times 33 = 330$ are genuine matches while the rest (13200) are impostor matches. The recognition accuracy of the system is evaluated based on the number of test samples whose minimum distance is achieved with a template from the same class. During verification, False Rejection Rate (FRR) measures the rate at which genuines are rejected as impostors while False Acceptance Rate (FAR) is the rate at which impostors are accepted as genuine. Receiver Operating Characteristic (ROC) curves, which plot the FAR against the FRR are used to obtain the Equal Error Rate (EER). This indicates the optimal operating point where both FRR and FAR are minimum. The Total Success Rate (TSR), which gives a measure of the verification performance of the system, is also obtained. This is given by

$$TSR (\%) = \left(1 - \frac{FA + FR}{\text{Total number of attempts}} \right) \times 100 . \quad (4)$$

where FA is the number of falsely accepted impostors and FR is the number of genuine palmprints rejected as impostors .

3.1 Recognition and Verification Results

Table 1 shows the recognition accuracy at different feature lengths per 8×8 and 16×16 block while varying the size of the input image from 128×128 to 32×32 pixels. The highest accuracy of 99.2% is achieved when using a 128×128 input image and 16×16 block size during DCT feature extraction with only the largest 25% of the coefficients used. As the image size is reduced, however, the recognition rate using the 16×16 block deteriorates much faster than the 8×8 block.

Table 2 shows the best TSR during verification for both block sizes. The values indicate the maximum TSR, which can occur at a different point from the EER because the number of impostor matches is much larger than the genuine matches. The value shown here for the 64×64 input image is obtained when the FAR and FRR are 0.1 % and 6.7 % respectively. The TSR therefore indicates the point where the system is achieving the most success, which is not necessarily the optimal operating point given by the EER.

Fig. 3 shows the ROC curves for the 2 block sizes at different input image sizes. It can be noted that the optimal results for verification are obtained when using a 128×128 input image and 16×16 block size for DCT, with an EER of 2.8 %. However, this value falls rapidly when the input image size is reduced so that the 8×8 block size performs much better at 64×64 and 32×32 image sizes.

Table 1. Recognition rate (%) for 16×16 and 8×8 block sizes at various image sizes

Block size	Features per block	Input image size		
		128×128	64×64	32×32
16×16	128	98.5	80.3	50.8
	64	99.2	78.8	47.7
	32	98.5	75.0	43.2
	2	93.2	62.1	25.8
8×8	32	98.5	93.2	75.8
	16	98.5	93.2	76.5
	2	93.9	86.3	62.1

Table 2. TSR for 16×16 and 8×8 block sizes

Block size	Features per block	TSR (%) for Input image size:		
		128×128	64×64	32×32
16×16	64	99.7	99.0	98.1
8×8	32	99.7	99.6	98.9

Because a reduction of the input image size by half increases the speed of processing by a factor of 4, this leads us to conclude that the 8×8 block size would be more suited for real-time applications. This is because halving the input image size increases the EER by only 0.4 % to 3.2 % if a 8×8 block size is used. However, when using the 16×16 block size, a reduction of the input image from the original size to 64×64 pixels causes the EER to increase from 2.8 % to 8.9 % as shown by the green line in Fig. 3 (b). At an input image size of 32×32 pixels, the 16×16 block size produces an EER of 18.4 % compared to 8.6 % for the 8×8 block size. This is shown by the red dotted line in Fig. 3.

The ROC curves in Fig. 3 also show that there is a direct correlation between the input image size and the block size chosen for DCT feature extraction. This can be explained by the fact that in a smaller input image, variations over a smaller local region will have a bigger effect on the overall discrimination between two images. By noting that the DC or fundamental value of the DCT transform is the average pixel value over the entire block, it is easy to see how the block size would be related to the input image size during discrimination.

3.2 Performance Comparison with Other Transform-Based Methods

Experimental results presented in Section 3 demonstrate the effectiveness of our method. To compare with global DCT feature extraction, we used the same input images and extracted half the DCT coefficients from the whole image and used their magnitudes as the features. The recognition accuracy from this was less than 50 %.

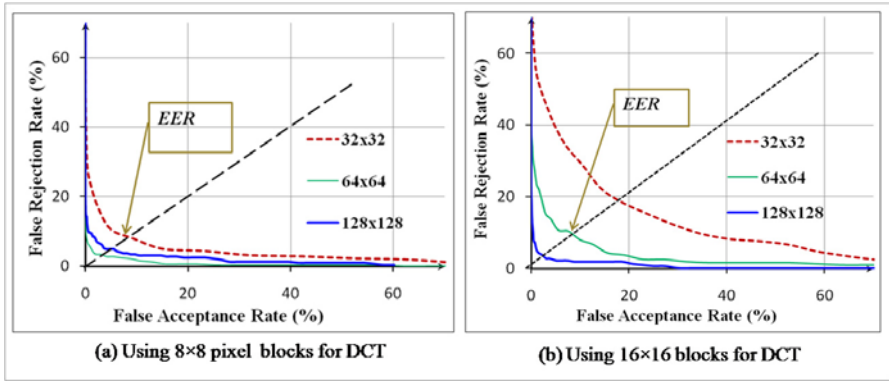


Fig. 3. The ROC curves for 3 input image sizes, (a) using an 8×8 block size and, (b) 16×16 block size during DCT feature extraction

Wen-xin, [17] used global DFT features on the same database and despite using two feature sets to represent direction and frequency, the maximum recognition accuracy obtained was 95.5 % and EER of 3.6 %. In previous work [17], we also used global DFT features but selected an optimal set of coefficients using a genetic algorithm. Using the same database a recognition rate of 98.9 % and EER of 2.5 % was achieved, but the length of the feature vector in this case was almost twice what it takes to obtain the maximum recognition accuracy here. Kumar and Shen [4] applied Eigenpalms on a database of 30 images of 3 palms and obtained a recognition rate of 98.7 %, while Zhang et al. [1] used a refined version of the same method on the PolyU palmprint database with an impressive recognition rate of 99.1 % and an EER of 1 %. In [16], global DCT for face and palmprint recognition is performed. Even after using the Fisher criterion to maximize the separability of a selected frequency band of DCT coefficients, the palmprint recognition rate achieved was 98.1 % and no value was reported for the EER.

4 Conclusions

In this paper, we proposed a holistic method of palmprint feature extraction and matching based on block-based DCT features and conducted various experiments to test the effectiveness of the method in recognition and verification using images from the PolyU Palmprint Database. These experiments show clearly that the use of local DCT features for palmprint recognition, when combined with careful preprocessing, is an effective alternative to other statistical or structural methods. Excellent recognition results are obtained when a 16×16 block size and a 128×128 pixel input image are used. In this case, using just 25 % of the largest DCT coefficients produces a recognition accuracy of 99.2 %, with 131 out of the 132 palms in the evaluation set correctly identified. The best EER of the method is 2.8 % while a maximum TSR of 99.7 % is achieved.

We also conducted experiments to determine the effect of the input image size on the recognition accuracy when the block size used for DCT is varied and showed that an 8×8 block size is more robust to reduction in the input image size, with the EER increasing from 3.2 % to 8.6 % when the input image size is reduced from 128×128 to 32×32 pixels. This is in stark contrast to the 16×16 block size, where the EER increases from 2.8 % to 18.4 % for the same reduction in input image size.

Other methods such as Eigenpalms and Fisherpalms require the use of large matrices to compute the basis vectors during training in order to transform the training set into the new KLT space. In the case of Fisherpalms, subsequent use of covariance matrices is also necessary in the calculation of within-class and between-class scatter. This makes such methods computationally expensive, whereas the proposed method can take advantage of available fast algorithms to evaluate the DCT coefficients in a very efficient way. The data-independence of the DCT also saves us from having to calculate the basis vectors every time new data is introduced.

As part of present and future work, an improved preprocessing method is investigated. The method used here for position normalization relies on the ability to properly segment the holes between the fingers. In many of the images in the database, these areas are partly occluded, which adversely affects the accuracy of normalization as approximate borders are used in such cases. Also, to reduce the effects of slight shifts in position and orientation, we shall consider adopting a method that uses overlapping blocks for DCT feature extraction where different degrees of overlap will be investigated.

References

1. Zhang, D.: *Palmpoint Authentication*. Kluwer Academic Publications, USA (2004)
2. Connie, T., Beng-Jin, A., Ong, M., Ling, D.: An Automated Palmpoint Recognition System. *Img. & Vision Comp.* 23, 501–515 (2005)
3. Zhang, D., Kong, W., You, J., Wong, M.: Online Palmpoint Identification. *IEEE Trans. on PAMI* 25(9), 1041–1050 (2003)
4. Kumar, A., Shen, H.C.: Recognition of Palmpoint Using Eigenpalms. In: *Proc. of CVPRIP, North Carolina* (2003)
5. Wu, X., Zhang, D., Wang, K.: Fisherpalms Based Palmpoint Recognition. *PRL* 24, 2829–2838 (2003)
6. Kong, W., Zhang, D., Li, W.: Palmpoint Feature Extraction Using 2D Gabor Filters. *PR* 36, 2339–2347 (2003)
7. Wen-xin, L., Zhang, D., Zhuo-qun, X.: Palmpoint Recognition Based on Fourier Transform. *J. Software* 13(5), 879–886 (2002)
8. Huang, D., Jia, W., Zhang, D.: Palmpoint Verification Based on Principal Lines. *PR* 41, 1316–1328 (2008)
9. Chen, J., Zhang, C., Rong, G.: Palmpoint Recognition Using Crease. In: *Proc. of ICIP*, pp. 234–237 (2001)
10. Wu, J., You, X., Tang, Y.Y., Cheung, W.: Palmpoint Identification Based on Non-Separable Wavelet Filter Banks. In: *Proc. of 19th IEEE ICPR*, pp. 1–4 (2008)

11. Kumar, A., Wong, D.C.M., Shen, H.C., Jain, A.K.: Personal Verification Using Palmprint and Hand Geometry Biometric. In: Kittler, J., Nixon, M.S. (eds.) AVBPA 2003. LNCS, vol. 2688, pp. 668–678. Springer, Heidelberg (2003)
12. Gonzalez, C., Woods, R.E.: Digital Image Processing, 2nd edn. Prentice Hall, NJ (2002)
13. Hafed, Z.M., Levine, M.D.: Face Recognition using the Discrete Cosine Transform. *IJC* 43(3), 167–188 (2001)
14. Eickeler, S., Muller, S., Rigoll, G.: Recognition of JPEG Compressed Images Based on Statistical Methods. *IVC* 18, 279–287 (2000)
15. Wallace, G.K.: The JPEG Still Picture Compression Standard. *IEEE Trans. on Cons. Elec.* 38(1), xviii–xxxiv (1992)
16. Jing, X., Zhang, D.: A Face and Palmprint Recognition Approach Based on Discriminant DCT Feature Extraction. *IEEE Trans. on SM & C-B* 34(6), 2405–2415 (2004)
17. Choge, H.K., Karungaru, S., Tsuge, S., Fukumi, M.: A DFT-Based Method of Feature Extraction for Palmprint Recognition. *IEEJ Trans. on EIS* 129(7), 1296–1304 (2009)
18. Ma, L., Tan, T., Wang, Y., Zhang, D.: Personal Identification Based on Iris Texture Analysis. *IEEE Trans. on PAMI* 25(12), 1519–1533 (2003)
19. PolyU Palmprint Database, <http://www.comp.polyu.edu.hk/~biometrics/>

Representative and Discriminant Feature Extraction Based on NMF for Emotion Recognition in Speech

Dami Kim^{1,3}, Soo-Young Lee^{1,2,3,*}, and Shun-ichi Amari³

¹ Brain Science Research Center and Department of Bio and Brain Engineering, KAIST

² Department of Electrical Engineering, KAIST

373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Korea (South)

³ Mathematical Neuroscience laboratory, Brain Science Institute, RIKEN,

2-2 Hirosawa, Wako-shi, Saitama 351-0198, Japan

kldami@neuron.kaist.ac.kr, sylee@neuron.kaist.ac.kr,

amari@brain.riken.jp

Abstract. For the emotion recognition in speech we had developed two feature extraction algorithms, which emphasize the subtle emotional differences while de-emphasizing the dominant linguistic components. The starting point is to extract 200 statistical features based on intensity and pitch time series, which are considered as the superset of necessary emotional features. Then, the first algorithm, rNMF (representative Non-negative Matrix Factorization), selects simple features best representing the complex NMF-based features. It first extracts a large set of complex almost-mutually-independent features by unsupervised learning and latter selects a small number of simple features for the classification tasks. The second algorithm, dNMF (discriminant NMF), extracts only the discriminate features by adding Fisher criterion as an additional constraint on the cost function of the standard NMF algorithm. Both algorithms demonstrate much better recognition rates even with only 20 features for the popular Berlin database.

Keywords: discriminant feature, feature extraction, feature selection, NMF, Fisher criterion.

1 Introduction

The recognition of emotion in speech is an important component for the efficient human-computer interactions. However, the primary information in human speech is linguistic, and the speaker-dependent and emotion-dependent information are minors. Therefore, the efficient features of the subtle emotional differences for the language-independent emotion recognition are difficult to extract and still under intensive study [1-5].

The popular speech features for emotional recognition include fundamental frequency (pitch), formants, MFCC, and energy. [1][2] Features based on manifold

* The research was conducted while S.Y. Lee was on sabbatical leave from KAIST and working at RIKEN BSI, and D. Kim was a visiting student at RIKEN BSI.

learning [3] and Teager energy operator [4] are also used. However, a large number of features are required for good recognition performance. For example, the AIBO team came out with a huge set of 200 features from intensity and pitch time series for excellent recognition performance. [5] However, these have many redundant features and therefore are not optimum.

The NMF (Non-negative Matrix Factorization) is an efficient feature extractor for non-negative data, and usually results in efficient features without redundancy. [6] However, the extracted features based on the unsupervised learning mainly represent the primary information and are not suitable for the subtle differences such as emotional contents in speech.

In this paper we present two algorithms based on NMF (Non-negative Matrix Factorization) which reduce redundancy among extracted features and also extract the subtle differences for efficient classification tasks. The first algorithm (rNMF) selects one “representative” feature from each of the NMF-extracted complex features, while the second algorithm (dNMF) extracts discriminant complex features by simultaneously maximizing Fisher criterion and minimizing the NMF cost function.

2 Baseline: AIBO Features

As the baseline we use the 200 features implemented by the AIBO team with excellent emotion recognition performance. [5] It is a bottom-up approach using an extensive feature set of low level statistics of prosodic parameters. As shown in Table 1, the features are based on time series of intensity, pitch, and MFCCs.

Table 1. 5x4x10 features used by AIBO team

Acoustic features	Derived series	Statistics	
Intensity	Minima	Mean	Variance
Lowpassed intensity	Maxima	Maximum	Minimum
Highpassed intensity	Duration between local extrema	Median	First quartile
Pitch	The series itself	Range	Third quartile
Derivatives of MFCC		Between quartile range	Mean absolute local derivatives

We use Berlin emotional speech database developed by the Technical University of Berlin [7]. Ten actors (five females and five males) generated ten German utterances (five short and five longer sentences) which could be interpretable in all seven emotions. The emotional states are neutral, happy, angry, sad, boredom, fear, and disgust. Totally 535 utterances were recorded with a sampling frequency of 48 kHz and later downsampled to 16 kHz.

The Support Vector Machine (SVM) is used as the classifier with one-vs.-the-other tactic, and the class with the maximum output value among the 7 SVMs is regarded as the final decision. We divided database into 5 sets to use the stratified 5-fold cross validation with 424 utterances for the training and 111 utterances for the testing.

3 Representative NMF (rNMF) Features

The mutual information (MI) between the class variable and the feature coefficient is a popular choice of selection criterion. However, the MI criterion is good only for statistically-independent features. As shown in Figure 1, with a given feature f_1 the feature f_3 adds more information to the class than the feature f_2 , which has a large MI with f_1 . Although both f_1 and f_2 may be selected by MI criterion, once you have f_1 , you do not need f_2 . Figure 2 shows the cross-correlation between the AIBO features. Obviously many features have large MI values with high correlation, and the MI is not a good criterion for the feature selection. The MI values for the 200 AIBO features in Figure 3 clearly show that several features from lowpassed intensity and pitch time series have large MI values but may be redundant.

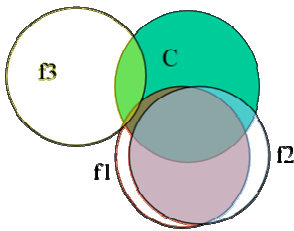


Fig. 1. MI between class and feature

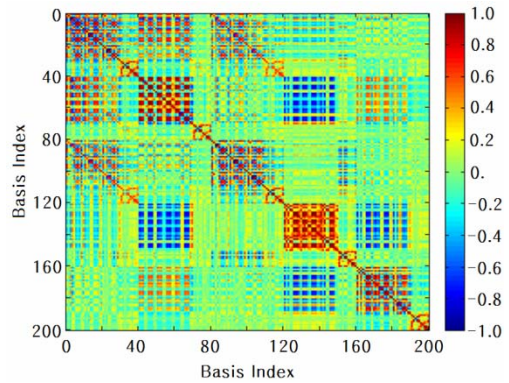


Fig. 2. Cross-correlation between AIBO feature

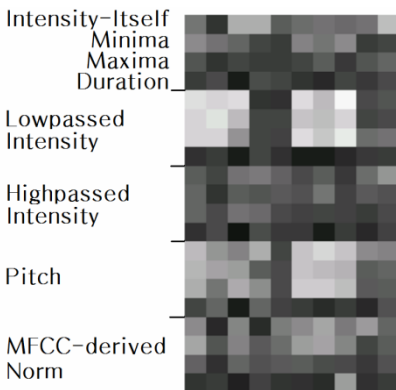


Fig. 3. MI between class and AIBO Features

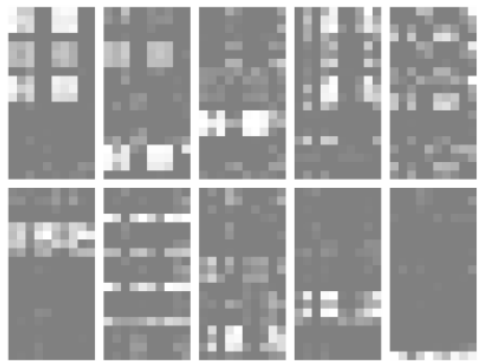


Fig. 4. 10 Complex features extracted by NMF

Features

Fortunately it is well known that several unsupervised feature extraction algorithms such as Independent Component Analysis (ICA) and NMF result in statistically-independent features. Due to the non-negative characteristics of the AIBO features the NMF algorithm is adopted here. Figure 4 shows the 10 features extracted by NMF. The low cross-correlation between NMF-based features is eminent in Figure 5.

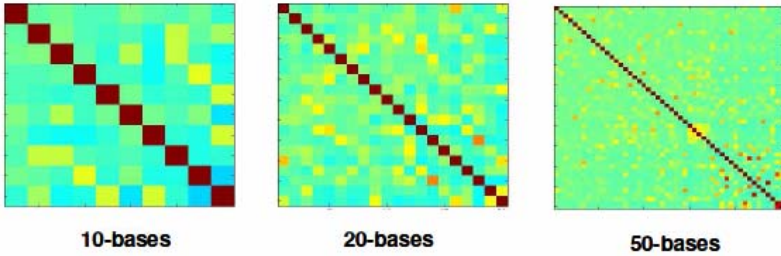


Fig. 5. Cross-correlation between NMF-based features

The NMF algorithm results in complex features, which are actually weighted linear combinations of original features. Although the extracted NMF features themselves may be used for the classification tasks, here we propose to select one “representative” original feature for each NMF feature. Since the NMF features have low cross-correlation, the “representative” original features may have low cross-correlations, too. In Figure 4 it is clear that the first, second and third NMF features mainly represent the intensity, MFCC, and pitch, respectively.

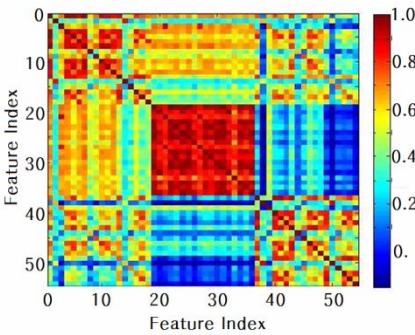


Fig. 6. Cross-correlation between 54 important original features for the first NMF feature in Figure 4



Fig. 7. Emotion recognition performance for 10 and 20 selected features by MI. The NMF-based features show the best performance.

Figure 6 shows the correlation matrix of 54 highly-weighted original features for the first NMF features in Figure 4. The mean, maximum, minimum, median, first quartile, and third quartile of the original, minima series, and maxima series of the intensity, lowpassed intensity, and highpassed intensity time series are important. In Figure 6, each successive 18 features came from the same acoustic series. The first

18 features are intensity related, while the next 18 features are derived from the lowpassed intensity. The last 18 features are derived from the highpassed intensity. Each 18 features are ordered as mean, maximum, minimum, median, first quartile, and third quartile of series itself, those of minima series, and those of maxima series. It is clear that the lowpassed intensity features are closely related to each other. They also have large MI with the class variable. Therefore, it is possible to select the “representative” original feature (rNMF feature) with the highest MI with the NMF-based feature. Table 2 summarizes the 10 “representative” features for the 10 NMF features. It is expected to have reasonably good recognition performance with these representative original features. Once one had selected these features, no NMF feature extraction is required for the test.

Table 2. Representative original features (rNMF features) of 10 NMF-based features

Basis	Acoustic Features	Derived Series	Statistics
Basis 1	Intensity	The Series Itself	Mean
Basis 2	Lowpassed Intensity	The Series Itself	Mean
Basis 3	Pitch	The Series Itself	Mean
Basis 4	Intensity	The Series Itself	Variance
Basis 5	Highpassed Intensity	The Series Itself	Mean of Absolute Derivative
Basis 6	Intensity	The Series Itself	First Quartile
Basis 7	Intensity	Duration Series	Mean of Absolute Derivative
Basis 8	Norm of MFCC derivative	The Series Itself	Variance
Basis 9	Pitch	The Series Itself	Variance
Basis 10	Norm of MFCC derivative	Duration Series	Mean

Figure 7 shows the emotion recognition performance with 10 and 20 selected features by MI criterion for original AIBO features, PCA features, NMF features, and the “representative” NMF (rNMF) features. The NMF-based features result in best recognition rates, of which only 10 and 20 features are comparable to that of all 200 AIBO features (71.2%). The 50 NMF features actually result in better recognition rate of 71.7%. The rNMF features which are just a small subset of the 200 AIBO feature result in much better recognition rates than those of AIBO features selected by MI criterion. The rNMF performance is only slightly inferior to those of NMF and PCA features. It clearly demonstrates that the “representative” feature selection from NMF-based features is an excellent choice.

4 Discriminant DNF (dNMF) Features

The discriminant features depend upon the classification task. For example the speech recognition relies on the classification of phonemes while neglecting speaker-dependent and emotion-dependent components. On the other hand the emotion recognition in speech needs amplify the subtle differences between emotional speeches while neglecting the phonemes and speaker-dependent components. The NMF and rNMF features are based on unsupervised learning without the knowledge on

the classification task, and naturally not optimum. Also, the unsupervised feature algorithms usually extract features with larger coefficient values, i.e. phoneme-related features in speech.

In this Section we describe the discriminant NMF (dNMF) which maximize discriminant performance during the NMF learning. It is a simultaneous feature extraction and selection algorithm, and may be regarded as an extension of NMF to incorporate Linear Discriminant Analysis (LDA) for multi-class problems.

For the dNMF one adds an additional cost function based on Fisher criterion for the discriminant power as

$$E = E_{NMF} + \lambda E_D$$

$$E_{NMF} = \|\mathbf{X} - \mathbf{WH}\|^2$$

$$E_D = -\frac{1}{2} \sum_{r=1}^R \log \left\{ \frac{1}{K} \sum_{k=1}^K (\mu_{rk} - \mu_r)^2 \right\} / \left\{ \frac{1}{N} \sum_{n=1}^N (H_m - \mu_{rk(n)})^2 \right\}$$

where λ is a weighting factor for the discriminant power, and \mathbf{X} , \mathbf{W} , and \mathbf{H} denote the original feature, NMF-based features, and their coefficient matrices, respectively. The mean coefficients of the k -th class and of all classes for the r -th feature are defined as

$$\mu_{rk} = \frac{1}{N_k} \sum_{n=1}^N H_m \delta_{kk(n)} \text{ and } \mu_r = \frac{1}{N} \sum_{n=1}^N H_m$$

Here $k(n)$ is the class of the n -th sample, and δ is the Cronecker delta. Also, R , K , N , and N_k denote the number of dNMF features, the number of classes, the number of training samples, and the number of samples in the k -th class, respectively. By minimizing the cost function E one is able to maximize between-class variance and minimize within-class variance of the feature coefficient, while still minimizing the representation error and maintaining the non-negativity.

A gradient-descent learning algorithm results in additional terms as

$$\frac{\partial E_D}{\partial H_m^{(d)}} = -\frac{(1/N_k(n))(\mu_{rk(n)} - \mu_r)}{\sum_{k=1}^K (\mu_{rk} - \mu_r)^2} - \frac{(H_m - \mu_{rk(n)})}{\sum_{n=1}^N (H_m - \mu_{rk(n)})^2}$$

Although the original learning rule in [6] utilizes multiplicative updates, we use a gradient-descent learning rule with an adaptive learning rate based on line-searching and progressive thresholding for the non-negativity. The resulting algorithm usually converges much faster than the multiplicative learning rule.

As shown in Figure 8, the difference of mean values between different classes for each feature coefficients $(\mu_{rk} - \mu_r)$ becomes larger during the NMF learning. In Figure 9 we show the emotion recognition rates with 5 to 20 features learnt by 3 different λ values, i.e. the weighting factor for the discriminant power. The best performance was achieved with only 15 features and $\lambda = 20$, and the recognition rate (69.9%) is much better than the best NMF performance (67.6%) with 20 features. Actually, it is not far from the 71.2% obtained for all the 200 AIBO features.

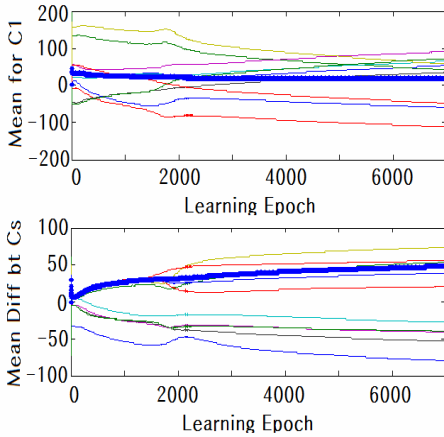


Fig. 8. Mean values of 10 dNMF features for the first class, and differences of mean values between the first class and the other classes. The differences increase during dNMF learning.

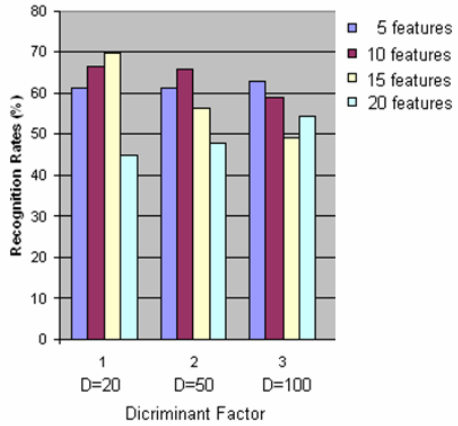


Fig. 9. Recognition rates of the dNMF with 5, 10, 15, and 20 features for 3 different values of discriminant factor $\lambda(=D)$. Fifteen features with $\lambda=20$ shows 69.9% which is close to the 71.2% with all 200 features.

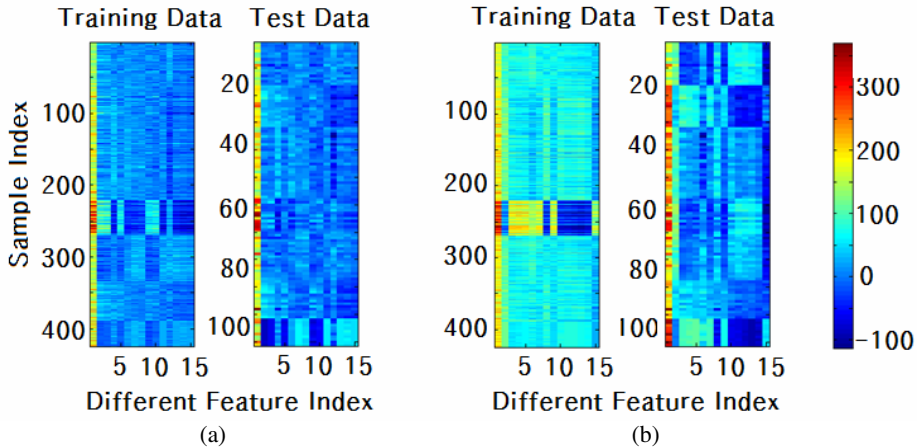


Fig. 10. Feature coefficients H_m for the training and test data with (a) $\lambda=20$ and (b) 50. In (b) they show different patterns for the training and test data, which is a symptom of overfitting.

With the discriminant factor $\lambda=20$ the recognition rates increase as the number of features increases up to 15, but falls down for 20 features. With the larger values of λ it becomes more serious. As shown in Figure 10, the feature coefficients have different patterns for the training and test data with the larger value of discriminant factor λ . The features are overfitted to the training data and do not generalize well for the test data. It is a common symptom of the supervised learning, which is usually avoided by having a validation dataset.

5 Conclusion and Future Research

In this paper we had demonstrated two efficient feature extraction algorithms for emotion recognition in speech. The representative NMF (rNMF) algorithm successfully select efficient raw features which represent the NMF-extracted complex features one-by-one. The discriminant NMF (dNMF) algorithm maximizes the discriminant power during the NMF feature extraction. Both algorithms result in much better classification performance than the simple feature selection based on MI. Especially, the dNMF algorithm results in excellent recognition rates with much smaller number of features, i.e., 15 out of 200 original features. Both algorithms are suitable to classify patterns based on subtle differences, not by the primary information.

In the future we will work on algorithms to overcome the overfitting problem of the dNMF. The dNMF cost function consists of the representation error E_{NMF} and the discriminant power E_D . The optimum features may be extracted based on the tradeoff between the two cost terms by adjusting the weighting factor λ . Also, an optimum value for the Fisher criterion may be imposed.

Acknowledgments. S.Y. Lee was supported by the Korea Research Foundation Grant (KRF-2008-013-D0091). D. Kim was also supported by Korea-Japan Collaboration Core Project on Neuroinformatics during her stay at RIKEN BSI.

References

1. Slaney, M., McRoberts, G.: Baby ears: a recognition system for affective vocalizations. *Speech Communications* 39, 367–384 (2003)
2. Lin, Y., Wei, G.: Speech emotion recognition based on HMM and SVM. In: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, August 2005, vol. 8, pp. 4898–4901 (2005)
3. You, M., Chen, C., Bu, J., Liu, J., Tao, J.: Emotional speech analysis on nonlinear manifold. In: 18th International Conference on Pattern Recognition, September 2006, vol. 3, pp. 91–94 (2006)
4. Zhou, G., Hansen, J.H.L., Kaiser, J.F.: Nonlinear feature based classification of speech under stress. *IEEE Transactions on Speech and Audio Processing* 9, 201–216 (2001)
5. Oudeyer, P.Y.: The production and recognition of emotions in speech: features and algorithms. *International Journal of Human-Computer Studies* 59(1), 157–183 (2003)
6. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 788–791 (1999)
7. Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W., Weiss, B.: A database of german emotional speech. In: Proceeding INTERSPEECH 2005, ISCA, pp. 1517–1520 (2005)

Improvement of the Neural Network Trees through Fine-Tuning of the Threshold of Each Internal Node

Hiroto Hayashi and Qiangfu Zhao

The University of Aizu, Aizuwakamatsu, Fukushima, Japan

Abstract. Neural network tree (NNTree) is a decision tree (DT) in which each internal node contains a neural network (NN). Experimental results show that the performance of the NNTrees is usually better than that of the traditional univariate DTs. In addition, the NNTrees are more usable than the single model fully connected NNs because their structures can be determined automatically in the induction process. Recently, we proposed an algorithm that can induce the NNTrees efficiently and effectively. In this paper, we propose to improve the performance of the NNTrees further through fine-tuning of the threshold of each internal node. Experimental results on several public databases show that, although the proposed method is very simple, the performance of the NNTrees can be improved in most cases, and in some cases, the improvement is even significant.

Keywords: Machine learning, pattern recognition, decision tree, neural network, multivariate decision tree.

1 Introduction

Neural networks (NNs) are a class of learning models analogous to the human brain. They often get good answers for solving non-linear problems, and have been applied successfully to many fields, such as image recognition, speech recognition, data mining, robot control, and so on. One drawback in using NNs is that determination of a proper network structure is usually difficult.

In our research, we have proposed a hybrid learning model called neural network tree (NNTree) [1]. An NNTree is a special decision tree (DT) with a small NN embedded in each internal node (see Fig. 1). The small NNs are used for local decisions, and the tree controls the whole decision making process. Usually, an NNTree is induced recursively. For the current node (start from the root), a small NN is added if the data assigned to this node are not pure enough (measured by the information gain ratio in this study). The small NN divides all data assigned to the current node into several groups. For each group, we do the same thing as above recursively. Because there is no trial and error, and the number

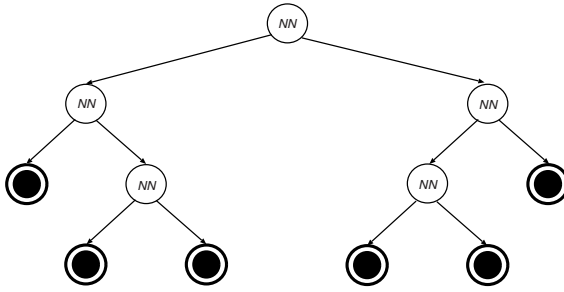


Fig. 1. An example of neural network trees

of small NNs needed is often proportional to the number of classes, the structure of the NNTree can be determined efficiently and automatically.

One bottleneck in using the NNTrees is the possible high cost for induction. Although the recursive induction process can finish in a very limited number of steps, finding the best NN in each internal node can be very time consuming. In fact, finding the best multivariate test function in each internal node is an NP-complete problem [2]. To solve the problem more efficiently, we have proposed an efficient algorithm based on a heuristic grouping strategy [3]. Using this algorithm, the NNTrees can be induced very quickly even for relatively large databases.

The purpose of this paper is to improve the stability of the induction algorithm. In fact, the existing algorithm is not bad. It can induce NNTrees with better generalization ability compared with the standard DT. It can also induce NNTrees that are comparable with the single model fully connected NNs. However, the existing algorithm is not very stable. For some databases, the performance of the NNTrees can be significantly worse than that of the single model fully connected NNs.

To improve the induction algorithm, we consider to fine-tune the threshold of each internal node of the NNTree once the NN embedded in this node is obtained. Here, the threshold of an internal nodes is defined as the bias of the output neuron of the NN. So far, we have fixed the threshold of output neurons to 0.5 (which is the medium value of the range of the output). This value may not follow the distribution of the outputs. In this paper, we propose to fine-tune the threshold based on the information gain ratio, so that the data can be divided more effectively using the same NN. Efficiency of this method is confirmed through experiments on several public databases.

The rest of the paper is organized as follows. Section 2 gives a brief review of DTs and NNTrees. Section 3 explains how to fine-tune the threshold of each internal node of the NNTree. Section 4 provides experimental results, and discusses the performance of the proposed method compared with existing method of inducing NNTrees. Section 5 is the conclusion.

2 Preliminaries

2.1 Definition of Decision Tree

Roughly speaking, a decision tree (DT) is a directed graph with no cycles. We usually draw a DT with the root at the top (see Fig. 1). Each node (except the root) has exactly one node above it, which is called its parent. The nodes directly below a node are called its children. A node is called a terminal node if it does not have any child. A node is called an internal node if it has at least one child. The node of a DT can be defined as a 5-tuple as follows:

$$node = \{I, F, Y, N, L\}$$

where I is a unique number assigned to each node, F is a test function that assigns a given input pattern to one of the children, Y is a set of pointers to the children, $N = |Y|$ is the number of children or the size of Y , and L is the class label of a terminal node (it is defined only for terminal node). For terminal nodes, F is not defined and Y is empty ($N=0$).

The process for recognizing an unknown pattern \mathbf{x} is as follows:

-
- Step 1: Set the root as the current node.
 - Step 2: If the current node is a terminal node, assign \mathbf{x} with the class label of this node, and stop; otherwise, find $i = F(\mathbf{x})$.
 - Step 3: Set the i -th child as the current node, and return to Step 2.
-

2.2 Induction of Decision Tree

To induce a DT, it is necessary to have a training set composing feature vectors and their class labels. The DT is induced by partitioning the feature space recursively. The induction process include three steps: splitting the nodes, finding terminal nodes, and assigning class labels to terminal nodes. The step of splitting nodes is the most significant and time consuming. To get a good DT, we should try to find a good test function for each internal node. Many criteria have been proposed for estimating the “goodness” of a test function [4], [5]. It is known that the efficiency of DTs is not affected greatly over a wide range of criteria [4]. In our study, we just adopt the information gain ratio (IGR), which is used in the well-known DT induction program C4.5 [5].

A test function $F(\mathbf{x})$ which maximizes the IGR decreases the entropy most for recognizing an unknown datum. Suppose S ($|S|$ is the size of S) is the set of data assigned to the current node, and n_i is the number of data belonging to the i -th class ($i = 1, 2, \dots, N_c$, and N_c is the number of classes), the entropy for recognizing an unknown datum is given by

$$info(S) = - \sum_{i=1}^{N_c} \frac{n_i}{|S|} \times \log_2\left(\frac{n_i}{|S|}\right) \quad (1)$$

Suppose S is divided into N subsets S_1, S_2, \dots, S_N by the test function F , the information gain is given as follows:

$$\text{gain}(F) = \text{info}(S) - \text{info}_F(S) \quad (2)$$

where

$$\text{info}_F(S) = \sum_{i=1}^N \frac{|S_i|}{|S|} \times \text{info}(S_i) \quad (3)$$

The IGR is defined by

$$\text{gain ratio}(F) = \text{gain}(F) \div \text{split info}(F) \quad (4)$$

where

$$\text{split info}(F) = - \sum_{i=1}^N \frac{|S_i|}{|S|} \times \log_2\left(\frac{|S_i|}{|S|}\right) \quad (5)$$

2.3 Definition of NNTrees

As mentioned previously, an NNTree is a kind of multi-variate decision tree, and each internal node contains a small NN. As the small NN, we use a small multilayer perceptron (MLP) here, although any kind of NNs can be adopted. The number of inputs and outputs of the NN correspond, respectively, to the dimensionality N_d of the feature space and the number N of child nodes. A major point in this research is to solve complex problems by embedding small NNs to the DT. Therefore, a small number (2, 4, or 6) is used as the number of hidden neurons as N_h in this paper.

Using an NNTree, any given example \mathbf{x} is recognized as follows:

-
- Step 1: Start from the root. This is the current node.
 - Step 2: If the current node is already a terminal node, assign its label to \mathbf{x} . Else find the next child node by using the following equation.

$$i = F(\mathbf{x}) = \arg \max_{1 \leq k \leq N} o_k \quad (6)$$

where o_k is the k -th output of the NN.

- Step 3: Set the i -th child as the current node, return to Step 2.
-

2.4 Induction of NNTrees

The overall process for inducing NNTrees is the same as that of C4.5. The difference is the method to generate the test function $F(\mathbf{x})$ in each internal node. For inducing NNTrees, it is not an efficient way to find the test function based on “generation and evaluation”. To find the test function more efficiently, we can

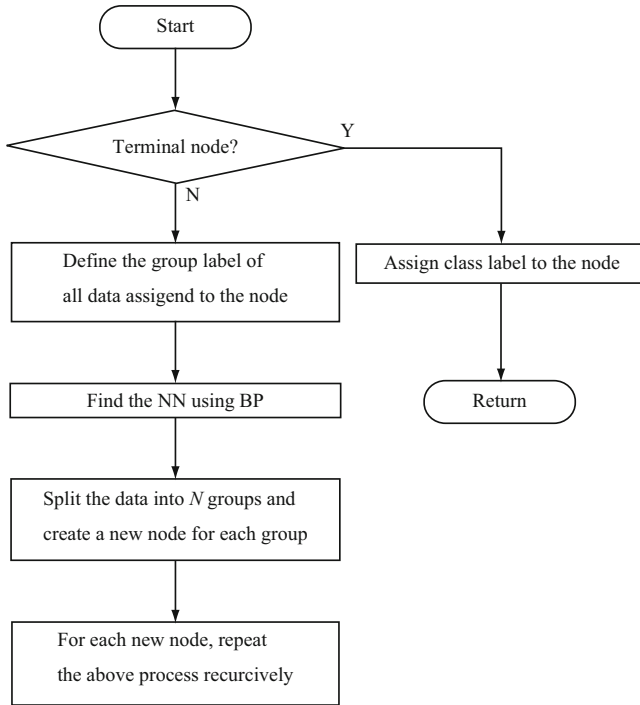


Fig. 2. Flowchart for inducing the NNTrees using the proposed method

define the teacher signals first using some heuristics [6]. Based on the teacher signals, we can find the NN test function quickly using the back-propagation (BP) algorithm [3]. This algorithm is much more efficient than the evolutionary algorithms we proposed before [1].

The flow-chart for inducing an NNTrees is given in Fig. 2. It can be explained as follows. The induction process is explained as follows.

-
- Step 1: Check if the current node is a terminal node. If examples assigned to this node belong to several classes, this node is an internal node, goto Step 3; otherwise, this node is a terminal node, goto Step2.
 - Step 2: Assign class label of the biggest classes to this node, and finish.
 - Step 3: Divide the examples assigned to the current node to N groups, and define the teacher signals for all data.
 - Step 4: Train the NN using the BP algorithm with the teacher signals defined in Step3.
 - Step 5: Split the examples into N groups using NN obtained at Step 4, and create a new node for each group.
 - Step 6: For each new node (child node), repeat the above process recursively.
-

Next, we explain how to divide examples into N subsets. The explanations of each step are as follows.

Suppose that we want to partition S (which is the set of examples assigned to the current node by the tree) into N sub-sets S_1, S_2, \dots, S_N , which are initially empty sets. For any given example $\mathbf{x} \in S$, repeat following process:

- Step 1: Get an unassigned data \mathbf{x} from S .
- Step 2: If there is a $\mathbf{y} \in S_i$, such that $label(\mathbf{y}) = label(\mathbf{x})$, assign \mathbf{x} to S_i ;
- Step 3: Else, if there is a S_i , such that $S_i = \Phi$, assign \mathbf{x} to S_i ;
- Step 4: Else, find \mathbf{y} , which is the nearest neighbor of \mathbf{x} in $\cup S_i$, and assign \mathbf{x} to the same sub-set as \mathbf{y} .

where \cup represents the union of sets, and Φ is the empty set.

The teacher signal of a data \mathbf{x} is the subset number to which it is assigned. That is, if \mathbf{x} is assigned to S_i , its teacher signal is i . The problem to design NNs as the test function changes to the supervised learning. In this research, we use the well known BP algorithm to train the NN test function.

3 Performance Improvement of the NN Trees through Fine-Tuning of Thresholds of Internal Nodes

3.1 Basic Concept

In this study, we study binary NN Trees only. For a binary NN Tree, one output neuron is enough to make the local decisions in each internal node (of course, the number of hidden neurons can be larger than or equal to two). So far, we have assumed that the threshold of each internal node is 0.5. That is, if the output of the NN is less than 0.5, visit the left child node; otherwise, visit the right child node.

After training the NN using the BP algorithm, the data assigned to the current node are projected by the NN to a 1-Dimensional space (Fig. 3). Most of the data are projected to the neighborhood of 0 or 1 because the square errors between the teacher signals and the actual outputs are minimized. If the training is successful, the data will be projected like Case 1 in Fig. 3. In general, however, the projected data may distribute like Case 2, or data of difference group might be mixed up like Case 3. In such cases, the recognition rate can be increased by fine-tuning the threshold.

3.2 Local Optimization of the Internal Node Thresholds Based on Information Gain Ratio

In this section, we explain how to decide the threshold. An important concept is information gain ratio (IGR) which has been explained in Chapter 2. The IGR is one of the well-known criteria for data division. The IGR becomes higher when a datum is assigned to the correct group. The best threshold is the value maximizing the IGR. To fine-tune the threshold, we follow the following steps:

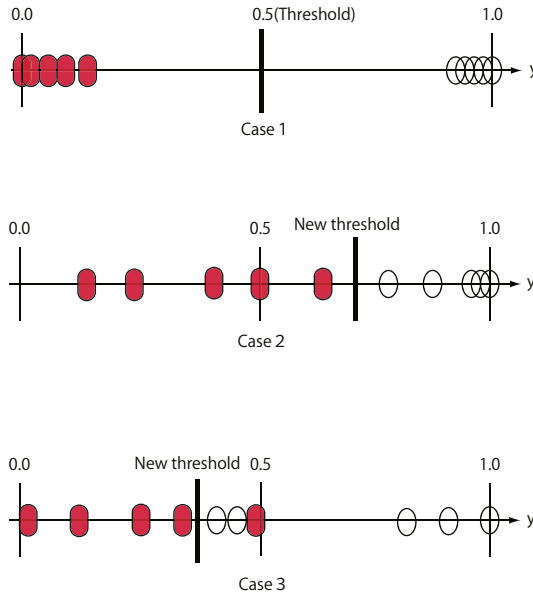


Fig. 3. Fine-tuning of threshold

-
- Step 1: Obtain the output values y_1, y_2, \dots, y_n of the NN for all data assigned to the current node.
 - Step 2: Sort the data according to the output values y_1, y_2, \dots, y_n .
 - Step 3: Calculate the average values $a_k = (y_k + y_{k+1})/2$, for $k = 1, 2, \dots, n-1$.
 - Step 4: Calculate $IGR(a_k)$, which is the information gain ratio corresponding to a_k , for $k = 1, 2, \dots, n-1$.
 - Step 5: The desired threshold is given by

$$T = \arg \max_k IGR(a_k).$$

The computational cost for obtaining the optimal threshold is $O(n)$. This increment of the cost can be ignored because it is much smaller than the cost of BP algorithm. In practice, the cost can be decreased by restricting the range of the threshold. For example, for the purpose of fine-tuning, T can be found in the range $[0.5 - \alpha, 0.5 + \alpha]$, rather than $[0, 1]$, where α is a small positive number.

We should also consider the timing of fine-tuning. There are two different ways. The first is to fine-tune the thresholds of all internal nodes after inducing the whole NNTree. The second is to fine-tune the threshold of each internal node once the NN in the current node is trained. Actually, the first way cannot induce good NNTrees because the data assigned to the child nodes can be changed when the threshold of the parent node is fine-tuned, and the child nodes may not work well any more. Therefore, we use the second way in this study.

4 Experimental Results

To verify the efficiency and efficacy of the proposed method, we conducted experiments with databases taken from the machine learning repository of the University of California at Irvine [7]. The databases used are adult, crx, dermatology, diabetes, ecoli, glass, isolet, pendigits, and soybean. Table 1 shows the parameters of the databases.

For each database, we conducted 5 trials of 10-fold cross validation (all together 50 runs). Each database was shuffled before each trial. The computer used in the experiments is Sun workstation: Sun Ultra20 M2 (the CPU is 2.2 GHz AMD Optron 1214, and the main memory is 1,024 MB).

We compare three approaches. The first one is the original induction algorithm. The second method is the original induction algorithm with threshold fine-tuning. The last method is a BP based single model full connected multi-layer perceptron (MLP).

We set experimental parameters of NNTrees as follows. First, for the BP algorithm, the learning rate is fixed to 0.5, and the maximum number of epochs for learning is 1,000. For the small NN in each internal node, the number of inputs of the NN is N_d . The number of hidden neurons is fixed to 4; and the number of output neuron is 1 because we consider only binary NNTrees here. The parameters of MLP are the same as previous one except the number of hidden neurons and output neurons. The number of hidden neurons of MLP is the sum of hidden neuron into each internal node of NNTrees. This is fair condition in the sense of system scale. A formula for the number of hidden neurons is (“number of internal node of NNTree” - 1) \times “number of hidden neurons of NN”. The number of output is the same as the number of the class of databases.

Table 2 shows the experimental results for all databases. The table contains “Test error” (the error rate for the test set); “Tree size” (the number of all nodes of the NNTree, including both terminal nodes and internal nodes); “Training time”; and the value of α for determining the search range of the threshold. For MLP, the number of hidden neurons is shown instead of the number of nodes. For each result, we have the average over 50 runs and the 95% confidence interval. Table 3 shows the result of t-test for the original induction algorithm and the

Table 1. Parameters of the Databases

	Number of examples (N_t)	Number of features (N_d)	Number of classes (N_c)
adult	48842	14	2
crx	690	15	2
dermatology	366	34	6
diabetes	760	2	8
ecoli	336	7	8
glass	214	9	6
ionosphere	351	34	2
isolet	7797	617	26
pendigits	10992	16	10
soybean	307	35	19

Table 2. Experimental results

	Method	Test error	Tree size	Training time	α
adult	Original	15.05±1.54	712.6±114.2	1018.56±129.72	-
	New	14.81±1.53	129.7±21.0	378.48±40.30	0.05
	MLP	42.31±7.10	(1424)	14865.93±7.78	-
crx	Original	16.87±1.98	14.5±1.0	2.86±0.13	-
	New	16.06±1.94	14.0±1.0	2.89±0.13	0.1
	MLP	15.80±2.03	(106)	4.29±0.08	-
dermatology	Original	3.83±1.41	16.0±0.6	0.08±0.02	-
	New	2.94±1.05	11.9±0.3	0.04±0.01	0.35
	MLP	2.67±0.97	(31)	0.10±0.02	-
diabetes	Original	25.76±2.55	26.1±2.5	2.87±0.18	-
	New	24.97±2.44	25.1±2.3	3.04±0.27	0.1
	MLP	26.18±2.48	(28)	6.07±0.10	-
ecoli	Original	15.64±2.49	15.6±1.1	0.78±0.04	-
	New	14.24±2.34	14.8±0.7	0.77±0.04	0.1
	MLP	13.70±2.27	(30)	2.22±0.06	-
glass	Original	34.00±3.46	21.9±1.5	0.68±0.05	-
	New	33.52±3.66	20.2±1.5	0.66±0.05	0.05
	MLP	33.52±3.33	(42)	1.97±0.06	-
ionosphere	Original	9.14±1.76	5.4±0.5	0.49±0.06	-
	New	7.89±1.74	5.5±0.5	0.40±0.06	0.45
	MLP	8.57±1.74	(9)	0.28±0.06	-
isolet	Original	10.65±2.07	75.4±2.8	268.96±20.85	-
	New	8.16±1.46	75.4±2.9	199.32±10.96	0.25
	MLP	10.32±1.46	(149)	8128.98±35.55	-
pendigits	Original	2.72±0.68	32.0±2.0	8.00±1.53	-
	New	2.37±0.62	29.7±1.5	4.03±1.11	0.45
	MLP	4.56±0.86	(63)	215.37±0.59	-
soybean	Original	18.00±2.41	53.5±1.8	0.48±0.05	-
	New	13.20±1.97	38.9±0.7	0.36±0.04	0.45
	MLP	11.33±2.06	(106)	16.28±0.96	-

proposed method. Table 4 also shows the result of t-test for the proposed method and MLP. The numerical number in this table is the number of databases which is significant difference against another.

First, we discuss about “Test error”. To compare the original induction method with the proposed method by t-test, the latter is better significantly for 4 databases (adult, isolet, pen, soybean). For other databases, there is no significant difference. The NNTree obtained by proposed method have been improved at recognition rate. To compare the original method with MLP, from the result of t-test, the original is better with significant difference for pen and adult database. The MLP is better with significant difference for soybean database. On the other hand, To compare the proposed method with MLP, the proposed one is better

Table 3. T-test for the original induction method and the new method

	Test Error	Tree size	Training time
Original	0	0	0
New	4	3	6
No difference	6	7	4

Table 4. T-test for the new method and method of NNTree and the new method MLP

	Test Error	Training time
New	3	9
MLP	0	1
No difference	7	0

significantly for 3 databases (adult, isolet, pen), and there are no significant difference for other databases. The recognition rate of proposed method have been more stable in the sense that NNTree obtained by the proposed method is better or comparable with that of MLP for all databases.

Next, let us consider the “Tree size”. To compare with the original method, the tree size of the proposed one is significant smaller for 3 databases (adult, dermatology, soybean), and there are no difference for other databases. Especially, the proposed method is more effective for “adult” database. The tree size can be reduced to about one fifth.

Finally, we discuss about the “Training time”. To compare with the original method by t-test, although the time for threshold fine-tuning is added, the proposed one is significant faster for 6 databases (adult, dermatology, ionosphere, isolet, pen, soybean), and there are no difference for other databases. This is mainly because that fine-tuning can reduce the number of errors in each internal node, and less “small” nodes will be produced during induction (By “small” node here we mean that only a small number of data are assigned to the node). Thus, the number of NN training is actually smaller if we use the proposed method.

5 Conclusion

In this paper, we have proposed an efficient method for inducing NNTrees through threshold fine-tuning. Experimental results show that, although the proposed method is very simple, it can get better results for all databases with no degrading performance, compared with the past proposed algorithm. In addition, the tree sizes and the training time can also be reduced to some extent using the proposed method.

References

1. Zhao, Q.F.: Evolutionary design of neural network tree - integration of decision tree, neural network and GA. In: Proc. IEEE Congress on Evolutionary Computation, pp. 240–244 (2001)
2. Murthy, S.K., Kasif, S., Salzberg, S.: A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research* 2(1), 1–32 (1994)
3. Hayashi, H., Zhao, Q.F.: A Fast Algorithm for Inducing Neural Network Trees. *IPSP Journal* 49(8), 2878–2889 (2008) (in Japanese)
4. Breiman, L., Friedman, J.H., Olshen, R.A., Stong, C.J.: *Classification and Regression Trees*. Wadsworth Pub. Co. (1984)
5. Quinlan, J.: *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco (1993)
6. Zhao, Q.F.: A New Method for Efficient Design of Neural Network Trees, Technical Report of IEICE, vol. PRMU2004-115, pp. 59–64 (2004)
7. Asuncion, A., Newman, D.J.: *UCI Machine Learning Repository*. University of California, School of Information and Computer Science, Irvine, CA (2007), <http://www.ics.uci.edu/~mllearn/MLRepository.html>

A Synthesis Method of Gene Networks Having Cyclic Expression Pattern Sequences by Network Learning

Yoshihiro Mori and Yasuaki Kuroe

Department of Information Science, Graduate School of Science and Technology,
Kyoto Institute of Technology, Matsugasaki, Sakyo-ku, Kyoto, Japan
{yoshihiro,kuroe}@kit.ac.jp

Abstract. Recently, the synthesis of gene networks having desired functions has become of interest to many researchers and several studies have been done. Synthesis methods of gene networks possessing desired expression pattern sequences are proposed. Periodic phenomena, e.g. circadian rhythm, are the important functions of cells. In this paper, we consider a synthesis problem of gene networks possessing the desired persistent cyclic expression pattern sequences. We have proposed the synthesis method of gene networks possessing the desired expression pattern sequences. Desired cyclic expression pattern sequences can be realized by using the synthesis method. But the behavior may not be persistent. We derive a sufficient condition such that the desired cyclic expression pattern sequences are persistent and propose a synthesis method realizing the persistent desired behavior by network learning.

1 Introduction

Investigating gene networks (GNs) is important for understanding mechanisms and functions of organisms and many researchers have been studied from various view points. Recently there have been increasing research interests in synthesizing GNs and several studies have been done. For example, [1] and [2] synthesize artificial GNs having oscillatory behavior. Those studies are motivated by two ways. One is that the synthesis of GNs could be the first step in controlling and monitoring biochemical processes in living cells. The other is that the synthesis of GNs is a complementary approach to investigating and understanding mechanisms of real GNs, that is to say, by synthesizing simple artificial networks and analyzing their behavior and functions, one can get some insights into functions of real GNs.

Recently, [3] and [4] propose a synthesis method of GNs having desired properties. In those studies the desired properties are given by expression pattern sequences (EPSs) which describe changes of expression levels of genes. There exist periodic phenomena in cells, e.g. circadian rhythm. These phenomena are caused by gene networks. Analyzing periodic behavior of gene networks could give some insight into understanding periodic phenomena of cells. In [4], we

show that the proposed method can synthesize GNs possessing cyclic EPSs. In order to make a GN possess a persistent cyclic EPS, its corresponding solution trajectory must be periodic. However, the synthesis method does not guarantee that the corresponding solution trajectory is periodic, which may cause that the realized EPS is cyclic, but not persistent.

In this paper, we propose a synthesis method of GNs possessing a desired persistent cyclic EPS. The proposed method realizes a periodic trajectory by assigning pass points of the trajectory. Constraint conditions such that a GN possesses a periodic solution trajectory passing through the points are derived. The synthesis problem is formulated as an optimization problem with the constraint conditions. An efficient algorithm to solve the optimization problem by network learning is derived.

2 Synthesis Problem

In this paper, we consider a continuous-time network model of GNs, which is given by the following differential equations⁵:

$$\begin{aligned} \dot{x}_i(t) &= -d_i x_i(t) + f_i(w_{i1}, w_{i2}, \dots, w_{im_i}, y_1(t), y_2(t), \dots, y_n(t)), & (1) \\ y_i(t) &= H(x_i(t)), \quad i = 1, 2, \dots, n, & (2) \end{aligned}$$

where n is the number of genes, $x_i(t)$ is a normalized expression quantity of the i th gene, $y_i(t) \in \{0, 1\}$ is a binary variable describing the on/off information of expression of the i th gene, that is, $y_i(t) = 1$ if the i th gene is expressed, $y_i(t) = 0$ if the i th gene is not expressed, $f_i : \{0, 1\}^n \rightarrow R$ is a nonlinear function describing interactions among genes, w_{ij} 's ($j = 1, 2, \dots, m_i$) are parameters of f_i , m_i is the number of the parameters of f_i , d_i denotes the degradation rate of the product of the i th gene and H is a threshold function:

$$H(x_i) = \begin{cases} 1 & \text{if } x_i \geq 0, \\ 0 & \text{if } x_i < 0. \end{cases} \tag{3}$$

This model is rewritten in the vector form:

$$\dot{x}(t) = -Dx(t) + f(w, y(t)), \quad y(t) = H(x(t)), \tag{4}$$

where $x^T = (x_1, x_2, \dots, x_n)$, $y^T = (y_1, y_2, \dots, y_n)$, $D = \text{diag} \{d_1, d_2, \dots, d_n\}$, $f^T = (f_1, f_2, \dots, f_n)$, $H^T(x) = (H(x_1), H(x_2), \dots, H(x_n))$, $w^T = (w_1^T, w_2^T, \dots, w_n^T)$ and $w_i^T = (w_{i1}, w_{i2}, \dots, w_{im_i})$. We call the vector y an expression pattern.

⁴ discussed the synthesis problem of GNs having the desired EPS: For a given EPS:

$$y^{*(0)} \rightarrow y^{*(1)} \rightarrow \dots \rightarrow y^{*(r)} \rightarrow \dots \rightarrow y^{*(p-1)} \rightarrow y^{*(p)}, \tag{5}$$

where p is the length of the sequence, synthesize a GN given by ⁴, which possesses a trajectory such that changes of the expression pattern $y(t)$ of the GN ⁴ become equal to the desired EPS given by ⁵.

In [4], it is shown that the proposed method can synthesize GNs having a cyclic EPS (5) in which $y^{*(0)} = y^{*(p)}$. In order to make a GN have a persistent cyclic EPS, its corresponding solution trajectory must be periodic. However, the synthesis method does not guarantee that the corresponding solution trajectory is periodic, which may cause that the realized EPS is cyclic, but not persistent.

The synthesis problem of GNs having the desired persistent cyclic EPS is formulated as:

Synthesis problem. For a given cyclic EPS:

$$y^{*(0)} \rightarrow y^{*(1)} \rightarrow \dots \rightarrow y^{*(r)} \rightarrow \dots \rightarrow y^{*(p-1)} \rightarrow y^{*(p)}, \quad y^{*(0)} = y^{*(p)}, \quad (6)$$

synthesize a GN (4) possessing a trajectory $\hat{x}(t)$ satisfying the following two conditions: (i) the solution trajectory $\hat{x}(t)$ is periodic ($\hat{x}(T + t) = \hat{x}(t)$), (ii) there exist time instants t_r 's ($r = 0, 1, \dots, p + 1$) such that $y^{*(r)} = H(\hat{x}(t))$ holds for the time interval $t_r \leq t < t_{r+1}$ ($r = 0, 1, \dots, p$).

Because it rarely happens that signs of multiple normalized expression quantities $x(t)$ change at the same time in real GNs, we assume that $\|y^{*(r+1)} - y^{*(r)}\|_2^2 = 1$, $r = 0, 1, \dots, p - 1$, where $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$ for $x \in R^n$. Note that this assumption implies that there exist i_r 's ($r = 0, 1, \dots, p - 1$) such that $y_{i_r}^{*(r)} \neq y_{i_r}^{*(r+1)}$ and $y_i^{*(r)} = y_i^{*(r+1)}$ for any $i \neq i_r$.

3 Synthesis Method

3.1 Problem Formulation as Optimization Problem

The synthesis problem of GNs having the desired EPS (5) can be formulated as an optimization problem with the condition (7) of the following theorem [4].

Theorem 1. For the given desired EPS (5), a GN (4) possesses the desired EPS (5) if it satisfies the conditions

$$y^{*(r+1)} = H(e(y^{*(r)})), \quad r = 0, 1, \dots, p - 1, \quad (7)$$

where $e(\hat{y}) := D^{-1}f(w, \hat{y})$.

To solve the synthesis problem, we must derive an additional condition corresponding to the periodicity of trajectories. Periodic solution trajectories corresponding the desired cyclic EPS given by (6) cross the boundaries $S_0, S_1, \dots, S_r, \dots, S_p$ in the order, where S_r is the boundary of the regions $\Omega_{y^{*(r)}}$ and $\Omega_{y^{*(r-1)}}$ ($r = 1, 2, \dots, p$), S_0 is identical to S_p due to $y^{*(0)} = y^{*(p)}$, and $\Omega_{\hat{y}}$ is a region in the space of $x(t)$ defined by $\Omega_{\hat{y}} := \{x \in R^n | \hat{y} = H(x)\}$. We take two steps for solving the synthesis problem; firstly, choose p points $x^{*(r)}$'s on the boundary S_r , $r = 0, 1, \dots, p$, where $x^{*(p)} = x^{*(0)}$ and secondly, synthesize a GN (4) having a solution trajectory $x(t)$ passing through the points $x^{*(r)}$, $r = 0, 1, \dots, p$, in the order.

We derive the condition such that a GN (4) has a periodic solution trajectory $x(t)$ passing through the points $x^{*(r)}$, $r = 0, 1, \dots, p$, in the order.

Theorem 2. For given desired EPS (6) and the points $x^{*(r)}$ on the boundary S_r , $r = 0, 1, \dots, p$, where $x^{*(0)} = x^{*(p)}$, a GN (4) possesses a cyclic trajectory passing through the points $x^{*(r)}$, $r = 0, 1, \dots, p$, in the order if it satisfies the conditions (7) and the conditions

$$x^{(r+1)} = x^{*(r+1)}, \quad r = 0, 1, \dots, p - 1, \tag{8}$$

where

$$x_i^{(r+1)} = e_i(y^{*(r)}) - \left(e_i(y^{*(r)}) - x_i^{*(r)} \right) \left(\frac{e_{i_r}(y^{*(r)})}{e_{i_r}(y^{*(r)}) - x_{i_r}^{*(r)}} \right)^{\frac{d_i}{d_{i_r}}}. \tag{9}$$

Proof. If a GN (4) satisfies the conditions (7), the trajectory of the GN (4) in $\Omega_{y^{*(r)}}$ starting from $x^{*(r)}$ is described as

$$x_i(t) = x_i^{*(r)} \exp(-d_i t) + e_i(y^{*(r)})(1 - \exp(-d_i t)), \tag{10}$$

$$i = 1, 2, \dots, n, \quad r = 0, 1, \dots, p - 1,$$

and we can see that the trajectory (10) crosses the surface S_{r+1} at $x^{(r+1)}$. Hence the trajectory starting from $x^{*(0)}$ is a cyclic trajectory passing through the points $x^{*(r+1)}$'s ($r = 0, 1, \dots, p - 1$) because the GN satisfies the conditions (8).

Values of the parameter vector w of the GN (4) satisfying the conditions (7) and (8) are not unique. We formulate the synthesis problem as an optimization problem whose constraints are (7) and (8) as follows.

$$\min_w J \text{ s. t. } y^{*(r+1)} = H(e(y^{*(r)})), \quad x^{(r+1)} = x^{*(r+1)}, \quad r = 0, 1, \dots, p - 1, \tag{11}$$

where J is a cost function depending on w , which represents a measure of the complexity of the GN (4). In this paper, we choose l_1 norm: $J = \sum |w_{ij}|$. A simpler GN (4) with smaller number of interactions could be obtained by the choice of J [4].

3.2 Learning Method for Synthesis Problem

It generally takes long computational time to solve differential equations. We must carefully solve the differential equations (4) because the GN model (4) is piecewise linear. To avoid these problems, we introduce the following discrete-time network:

$$x[k + 1] = D^{-1} f(w, y[k]), \quad y[k] = H(x[k]), \quad x[0] = x_0. \tag{12}$$

Let $x[k, x_0]$ and $y[k, x_0]$ be the solutions of the difference equations (12). Define $x^{[k, x_0]}$ as

$$x_i^{[k+1, x_0]} = x_i[k + 1, x_0] - \left(x_i[k + 1, x_0] - x_i^{*(k)} \right) \left(\frac{x_{i_k}[k + 1, x_0]}{x_{i_k}[k + 1, x_0] - x_{i_k}^{*(k)}} \right)^{\frac{d_i}{d_{i_k}}},$$

$$k = 0, 1, \dots, p - 1, \tag{13}$$

and $x_i^{[0,x_0]} = x_i^{*(0)}$. Using the discrete-time network (12), we reduce the optimization problem (11) to an optimization problem:

$$\min_w \hat{J}, \tag{14}$$

where $\hat{J} = \alpha J + \beta J_1 + \gamma J_2$, α , β and γ are weighting coefficients,

$$J_1 = \frac{1}{2} \sum_{k=1}^p \|y[k, x_0] - y^{*(k)}\|_2^2, \tag{15}$$

$$J_2 = \frac{1}{2} \sum_{k=1}^p \|x^{[k,x_0]} - x^{*(k)}\|_2^2, \tag{16}$$

and $x_0 \in \Omega_{y^{*(0)}}$. If $\hat{J} = 0$ for w^* , then $y[k, x_0]$ becomes equal to $y^{*(k)}$ and $x^{[k,x_0]}$ becomes equal to $x^{*(k)}$ for $k = 1, 2, \dots, p$. These imply that $x[k + 1, x_0] = e(y^{*(k)})$, $y^{*(k+1)} = H(e(y^{*(k)}))$, and $x^{(k+1)} = x^{*(k+1)}$ for $k = 0, 1, \dots, p - 1$. Hence, w^* satisfies the conditions (7) and (8).

The optimization problem (14) can be solved by network learning with the gradient based methods. To calculate the gradient of \hat{J} , the threshold function H in the discrete-time network (12) is replaced by a smooth function S which can closely approximate to H , then we introduce the discrete-time network:

$$x[k + 1] = D^{-1}f(w, y[k]), \quad y[k] = S(x[k]), \tag{17}$$

where $S(x) = (S(x_1), S(x_2), \dots, S(x_n))^T$. The function J is non-smooth. We define the gradient $\partial J / \partial w_{ij}$ of the function J as

$$\frac{\partial J}{\partial w_{ij}} = \begin{cases} 1 & \text{if } w_{ij} > 0 \\ 0 & \text{if } w_{ij} = 0 \\ -1 & \text{if } w_{ij} < 0. \end{cases} \tag{18}$$

Remark 1. From the definition (13), $x^{[k+1,x_0]}$ becomes a complex number if $x_{i_k}[k + 1, x_0] / (x_{i_k}[k + 1, x_0] - x_{i_k}^{*(k)})$ is negative. However, if the parameter vector w satisfies the condition (7), $x^{[k+1,x_0]}$ becomes the cross point $x^{(k+1)}$. This fact implies that $x^{[k+1,x_0]}$'s ($k = 0, 1, \dots, p - 1$) are real number. In order avoid the problem that $x^{[k+1,x_0]}$ becomes a complex number, we solve the optimization problem (14) by network learning with \hat{w} as initial values of w where \hat{w} is a solution of the optimization problem (14) with $\gamma = 0$. Note that \hat{w} satisfies the condition (7).

Remark 2. If the interaction functions are defined as

$$f_i(a_i, y) = a^{(i)} + \sum_{j=1}^n a_j^{(i)} y_j + \sum_{j=1}^{n-1} \sum_{k=j+1}^n a_{jk}^{(i)} y_j y_k + \dots + a_{12\dots n}^{(i)} y_1 \dots y_n, \tag{19}$$

the discrete-time network (17) with the interaction functions (19) is equivalent to a class of Recurrent High-Order Neural Networks(RHONNs). Hence the synthesis problem is solved by learning of RHONNs. The algorithm to compute

gradient $\partial J_1/\partial w_{ij}$ and $\partial J_2/\partial w_{ij}$ can be obtained based on the sensitivity analysis method by using adjoint equations or sensitivity equations [6].

Remark 3. GNs (4) having two or more desired persistent cyclic EPSs can be also synthesized by using the proposed synthesis method. The objective function \hat{J} in the optimization problem (14) is replaced with

$$\hat{J} = \alpha J + \beta \sum_{l=1}^q J_{1,l} + \gamma \sum_{l=1}^q J_{2,l}, \tag{20}$$

where

$$J_{1,l} = \frac{1}{2} \sum_{k=1}^{p_l} \|y[k, x_{0,l}] - y^{*(k,l)}\|_2^2, \tag{21}$$

$$J_{2,l} = \frac{1}{2} \sum_{k=1}^{p_l} \|x^{[k,x_{0,l}]} - x^{*(k,l)}\|_2^2, \tag{22}$$

$x_{0,l} \in \Omega_{y^{*(0,l)}}$, $y^{*(k,l)}$'s ($k = 1, 2, \dots, p_l$) are the expression patterns consisting of the l th EPS, p_l is the length of l th EPS, $x^{*(k,l)}$ is the assigned point on the boundary $S_{k,l}$ of $\Omega_{y^{*(k,l)}}$ and $\Omega_{y^{*(k-1,l)}}$, and $x_{0,l}$'s ($l = 1, 2, \dots, q$) are initial state of $x[k]$.

4 Numerical Experiments

We have carried out experiments in order to illustrate the performance of the synthesis method. In these experiment, we use the interaction functions (19) and a sigmoidal function $S(x) = 1/(1 + \exp(-7x))$ for a smooth function S , which approximates the threshold function H . We let the parameters d_i 's of genes be $d_i = 1.0$. The weighting coefficients α , β and γ in the cost function \hat{J} are determined by trial and error.

4.1 Realization of a Cyclic Expression Pattern Sequence

Let a desired persistent cyclic EPS be given as:

$$\begin{aligned} (0, 0, 0, 0, 0)^T &\rightarrow (1, 0, 0, 0, 0)^T \rightarrow (1, 1, 0, 0, 0)^T \rightarrow (1, 1, 1, 0, 0)^T \\ &\rightarrow (1, 1, 1, 1, 0)^T \rightarrow (1, 1, 1, 1, 1)^T \rightarrow (0, 1, 1, 1, 1)^T \rightarrow (0, 0, 1, 1, 1)^T \\ &\rightarrow (0, 0, 0, 1, 1)^T \rightarrow (0, 0, 0, 0, 1)^T \rightarrow (0, 0, 0, 0, 0)^T. \end{aligned} \tag{23}$$

A GN (4) consisting of 5 genes is synthesized because the expression patterns have 5 elements. We choose the points $x^{*(r)}$ on the boundaries S_r 's ($r = 0, 1, \dots, 10$) as

$$x^{*(0)} = (-1.0, -2.0, -2.0, -1.0, 0.0)^T, \quad x^{*(1)} = (0.0, -1.0, -2.0, -2.0, -1.0)^T,$$

$$\begin{aligned}
 x^{*(2)} &= (1.0, 0.0, -1.0, -2.0, -2.0)^T, & x^{*(3)} &= (2.0, 1.0, 0.0, -1.0, -2.0)^T, \\
 x^{*(4)} &= (2.0, 2.0, 1.0, 0.0, -1.0)^T, & x^{*(5)} &= (1.0, 2.0, 2.0, 1.0, 0.0)^T, \\
 x^{*(6)} &= (0.0, 1.0, 2.0, 2.0, 1.0)^T, & x^{*(7)} &= (-1.0, 0.0, 1.0, 2.0, 2.0)^T, \\
 x^{*(8)} &= (-2.0, -1.0, 0.0, 1.0, 2.0)^T, & x^{*(9)} &= (-2.0, -2.0, -1.0, 0.0, 1.0)^T, \\
 x^{*(10)} &= x^{*(0)}, & &
 \end{aligned}
 \tag{24}$$

and set the weighting coefficients α, β and γ in the cost function \hat{J} as $\alpha = 0.001, \beta = 1.0$ and $\gamma = 10.0$ and initial states x_0 of $x[k]$ as $x_0 = (-1.0, -1.0, -1.0, -1.0, -1.0)^T$, respectively. Applying the synthesis method, a parameter vector w of a GN (4) having a trajectory passing through the points $x^{*(r)}$'s ($r = 0, 1, \dots, 10$) is obtained. An example of simulation results of the synthesized GN (4) is shown in Fig. 1, where initial state $x(0)$ is the same as $x^{*(0)}$ in (24). The numbers

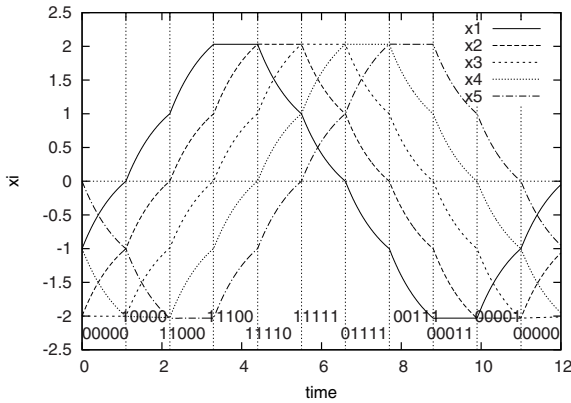


Fig. 1. Simulation result of the obtained GN : realization of a cyclic pattern sequence

placed at the bottom of Fig. 1 represent the expression patterns of the synthesized GN. Vertical dashed lines show boundaries where the expression pattern $y(t)$ of the GN changes. It is observed that the obtained GN (4) has the desired EPS (23) with its corresponding periodic trajectory passing through the points (24). It is concluded that the obtained GN (4) possesses the desired persistent EPS (23).

4.2 Realization of Two Cyclic Expression Pattern Sequences

Let two desired persistent cyclic EPS be given as:

$$\begin{aligned}
 &(1, 0, 0, 0, 0)^T \rightarrow (1, 1, 0, 0, 0)^T \rightarrow (0, 1, 0, 0, 0)^T \rightarrow (0, 1, 1, 0, 0)^T \\
 &\rightarrow (0, 0, 1, 0, 0)^T \rightarrow (0, 0, 1, 1, 0)^T \rightarrow (0, 0, 0, 1, 0)^T \rightarrow (0, 0, 0, 1, 1)^T \\
 &\rightarrow (0, 0, 0, 0, 1)^T \rightarrow (1, 0, 0, 0, 1)^T \rightarrow (1, 0, 0, 0, 0)^T
 \end{aligned}
 \tag{25}$$

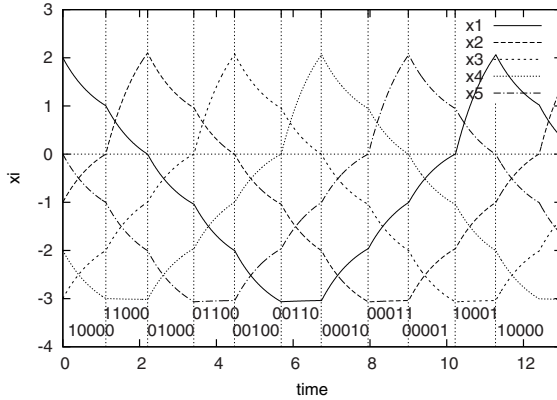


Fig. 2. Simulation result of the obtained GN : realization of two cyclic pattern sequence

and

$$\begin{aligned}
 &(1, 1, 1, 0, 0)^T \rightarrow (1, 1, 1, 1, 0)^T \rightarrow (0, 1, 1, 1, 0)^T \rightarrow (0, 1, 1, 1, 1)^T \\
 &\rightarrow (0, 0, 1, 1, 1)^T \rightarrow (1, 0, 1, 1, 1)^T \rightarrow (1, 0, 0, 1, 1)^T \rightarrow (1, 1, 0, 1, 1)^T \\
 &\rightarrow (1, 1, 0, 0, 1)^T \rightarrow (1, 1, 1, 0, 1)^T \rightarrow (1, 1, 1, 0, 0)^T.
 \end{aligned} \tag{26}$$

A GN (4) consisting of 5 genes is synthesized because the expression patterns have 5 elements. We choose the points $x^{*(r,l)}$'s on the boundaries $S_{r,l}$'s ($r = 0, 1, \dots, 10, l = 1, 2$) for the desired EPS (25) and (26) as

$$\begin{aligned}
 x^{*(0,1)} &= (2.0, -1.0, -3.0, -2.0, 0.0)^T, & x^{*(1,1)} &= (1.0, 0.0, -2.0, -3.0, -1.0)^T, \\
 x^{*(2,1)} &= (0.0, 2.0, -1.0, -3.0, -2.0)^T, & x^{*(3,1)} &= (-1.0, 1.0, 0.0, -2.0, -3.0)^T, \\
 x^{*(4,1)} &= (-2.0, 0.0, 2.0, -1.0, -3.0)^T, & x^{*(5,1)} &= (-3.0, -1.0, 1.0, 0.0, -2.0)^T, \\
 x^{*(6,1)} &= (-3.0, -2.0, 0.0, 2.0, -1.0)^T, & x^{*(7,1)} &= (-2.0, -3.0, -1.0, 1.0, 0.0)^T, \\
 x^{*(8,1)} &= (-1.0, -3.0, -2.0, 0.0, 2.0)^T, & x^{*(9,1)} &= (0.0, -2.0, -3.0, -1.0, 1.0)^T, \\
 x^{*(10,1)} &= x^{*(0,1)},
 \end{aligned} \tag{27}$$

and

$$\begin{aligned}
 x^{*(0,2)} &= (2.0, 3.0, 1.0, -1.0, 0.0)^T, & x^{*(1,2)} &= (1.0, 3.0, 2.0, 0.0, -2.0)^T, \\
 x^{*(2,2)} &= (0.0, 2.0, 3.0, 1.0, -1.0)^T, & x^{*(3,2)} &= (-2.0, 1.0, 3.0, 2.0, 0.0)^T, \\
 x^{*(4,2)} &= (-1.0, 0.0, 2.0, 3.0, 1.0)^T, & x^{*(5,2)} &= (0.0, -2.0, 1.0, 3.0, 2.0)^T, \\
 x^{*(6,2)} &= (1.0, -1.0, 0.0, 2.0, 3.0)^T, & x^{*(7,2)} &= (2.0, 0.0, -2.0, 1.0, 3.0)^T, \\
 x^{*(8,2)} &= (3.0, 1.0, -1.0, 0.0, 2.0)^T, & x^{*(9,2)} &= (3.0, 2.0, 0.0, -2.0, 1.0)^T, \\
 x^{*(10,2)} &= x^{*(0,2)},
 \end{aligned} \tag{28}$$

respectively. We set the weighting coefficients α, β and γ in the cost function \hat{J} as $\alpha=0.0001, \beta = 1.0$ and $\gamma = 1.0$, and initial states $x_{0,l}$ and $x_{0,2}$ of $x[k]$

as $x_{0,1} = (1.0, -1.0, -1.0, -1.0, -1.0)^T$ and $x_{0,2} = (1.0, 1.0, 1.0, -1.0, -1.0)^T$, respectively. Applying the synthesis method, a parameter vector w of a GN (4) having two periodic trajectories is obtained. One trajectory passes through the points (27) and the other passes through the points (28). From simulations of the GN by using the obtained parameters, it is confirmed that the GN (4) has the desired cyclic EPSs (25) and (26) with their corresponding periodic trajectories passing through the desired points (27) and (28), respectively. An example of simulation results of the synthesized GN (4) is shown in Fig. 2, where initial state $x(0)$ is the same as $x^{*(0,1)}$ in (27). It is observed that the obtained GN (4) has the desired EPS (25) with its corresponding periodic trajectory passing through the points (27). It is concluded that the obtained GN (4) possesses the desired persistent EPSs (25) and (26).

5 Conclusion

There exist periodic phenomena in cells, e.g. circadian rhythm. These phenomena are caused by gene networks. In this paper, we proposed a synthesis method of gene networks possessing the desired persistent cyclic expression pattern sequence. The proposed method realize its corresponding periodic solution trajectory. To solve the synthesis problem, we introduced the discrete-time network having the equivalent behavior to the expression pattern sequence of gene network. The synthesis problem was reduced to an optimization problem by using the discrete-time network. The efficient algorithm to solve the optimization problem was derived.

References

1. Elowitz, M.B., Leibler, S.: A synthetic oscillatory network of transcriptional regulators. *Nature* 403, 335–338 (2000)
2. Fung, E., Wong, W.W., Suen, J.K., Bulter, T., Lee, S., Liao, J.C.: A synthetic gene-metabolic oscillator. *Nature* 435, 118–122 (2005)
3. Ichinose, N., Aihara, K.: A gene network model and its design. In: 15th Workshop on Circuit and Systems, pp. 589–593 (2002) (in Japanese)
4. Mori, Y., Kuroe, Y., Mori, T.: A synthesis method of gene networks based on gene expression by network learning. In: SICE-ICASE International Joint Conference, pp. 4545–4550 (2006)
5. Glass, L.: Classification of biological networks by their qualitative dynamics. *Journal of Theoretical Biology* 54, 85–107 (1975)
6. Kuroe, Y., Ikeda, H., Mori, T.: Identification of nonlinear dynamical systems by recurrent high-order neural networks. In: 1997 IEEE International Conference on Systems Man and Cybernetics, vol. 1, pp. 70–75 (1997)

Gender Identification from Thai Speech Signal Using a Neural Network

Rong Phoophuangpairroj, Sukanya Phongsuphap, and Supachai Tangwongsan

Faculty of Information and Communication Technology, Mahidol University,
999 Phuttamonthon 4 Road, Salaya, Nakhonpathom 73170, Thailand
g4536827@student.mahidol.ac.th,
ccsps@mahidol.ac.th,
ccstw@mahidol.ac.th

Abstract. This paper proposes a method for identifying a gender by using a Thai spoken syllable with the Average Magnitude Difference Function (AMDF) and a neural network (NN). The AMDF is applied to extracting pitch contour from a syllable. Then the NN uses the pitch contour to identify a gender. Experiments are carried out to evaluate the effects of Thai tones and syllable parts on the gender classification performance. By using a whole syllable, the average correct classification rate of 98.5% is achieved. While using parts of a syllable, the first half part gives the highest accuracy of 99.5%, followed by the middle and the last parts with the accuracies of 96.5% and 95.5%, respectively. The results indicate that the proposed method using pitch contour from any tones of the first half of a Thai spoken syllable or a whole Thai spoken syllable with the NN is efficient for identifying a gender.

Keywords: Gender identification, Thai speech signal, Thai syllables, Thai tones, Neural network.

1 Introduction

The speech signal has been studied by researchers for several applications such as speech recognition, speaker identification, and robotic interaction. It is well-known that speech signal not only conveys the message but also a lot of information about the speaker himself such as a gender. The gender appears to be the important factor related to physiological differences that create speech variability [1][2]. A speaker's gender can be one of the variabilities adversely affecting the speech recognizer's accuracy and apparently, separating speakers can be considered as an important way of improving a speech recognizer's performance [3]. In speech recognition and speaker identification applications, the studies in the literature show that gender-dependent models are more accurate than gender-independent ones [4] and these applications would be simpler, if we could recognize a speaker's gender [5]. In robotic applications, the robots can interact with users by providing suitable services to females and males according to their gender information [6]. The gender classification can also be used to create more security for the places that allow only for females

or males. In this work, we aim at Thai speech recognition applications. In the Thai speech recognition, there is evidence that gender-dependent acoustic models can achieve higher recognition results than gender-independent acoustic models [7]. However, using gender-dependent acoustic models, the gender of a speaker has to be classified beforehand. Therefore, if the gender of speakers can be recognized accurately from the beginning part of speech or a spoken syllable, the results of Thai speech recognition applications can be boosted.

The pitch frequency and gender are studied in languages [8][9]. However, the patterns and the number of pitch contours differ from language to language. The average pitch frequency is used for recognizing genders, but the accuracy is still unsatisfactory [10]. We think that a syllable unit contains rather rich pitch information, which should be represented by the pitch frequency contour not just the average pitch frequency. Therefore, the pitch contour of Thai syllables and parts of them should be studied more for the gender classification task. In this work, the gender classification based on a Thai spoken syllable and parts of the syllable is investigated. The paper is organized as follows. Section 2 describes Thai speech signal analysis of Thai syllables with different tones. Section 3 explains the proposed method to identify the gender. Section 4 carries out several experiments of the gender classification using the whole and half parts of a Thai syllable. Finally, section 5 gives conclusions.

2 Thai Speech Signal Analysis

Thai is a tonal language that has five different tones: mid, low, falling, high, and rising, respectively. A base syllable with a different tone always means different things. The shapes of pitch contours of the syllables having the same tones are quite similar [11] and rather independent to Thai vowels [12]. Fig. 1 shows the fundamental frequencies (F0) or pitch contour of Thai syllables with five different tones when they are spoken in isolation by a female speaker and a male speaker.

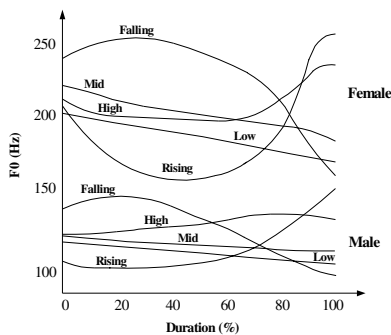


Fig. 1. Fundamental frequencies of five Thai tones of a female and a male

Fig. 1 reveals that the levels of the pitch frequencies of a female speaker are higher than those obtained from a male speaker. Therefore, in the Thai language, a syllable can be a good source of gender information and it is possible to use pitch contour

extracted from a Thai syllable to classify speakers' gender. The pitch contours of Thai syllables from several speakers should be considered. Additionally, to comprehensively understand the effect of Thai tones on gender classification, the use of a mid-tone through rising-tone syllable and parts of a syllable with a gender classification method should be investigated.

3 The Proposed Gender Classification Method

The proposed gender classification method comprises three stages, which are pitch frequency extraction using the Average Magnitude Difference Function (AMDF), pitch feature representation for a neural network, and gender classification by a neural network classifier, as shown in Fig. 2.

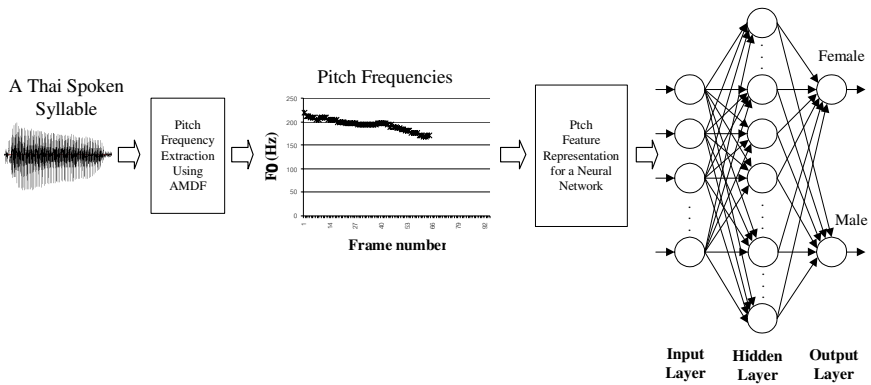


Fig. 2. Gender classification based on a Thai syllable using a neural network

In the first stage, the acoustic pitch frequencies (F0) are extracted from a speech signal of a Thai spoken syllable. As a consequence, the pitch contour consisting of a number of pitch frequencies is obtained from the signal. The numbers of pitch frequencies of speech signals are not equal because the duration of syllables spoken by speakers is not equivalent. In the second stage, the pitch frequencies are selected or generated to fit the number of the neural network inputs. In the last stage, the gender is identified using a neural network classifier. The details of each stage are explained in the following subsections.

3.1 Pitch Frequency Extraction Using AMDF

Acoustic feature is one of the most important factors in classification using speech. In this research, pitch frequencies (fundamental frequencies) extracted from speech signal are used to determine the gender of speakers. There is evidence that voice speech is quasi-periodic [13]. The quasi-periodic signal obtained from a voice part of a Thai spoken syllable is illustrated in Fig. 3.

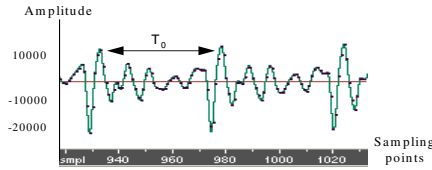


Fig. 3. A quasi-periodic signal obtained from a Thai spoken syllable

To calculate the fundamental frequency, the AMDF is used. The AMDF is a measure of periodicity of speech waveform. The function is expected to have a minimum when the shift variable k in the following equation equals the waveform period T_0 of a quasi-periodic signal of a syllable $x(n)$ of length K .

$$a(k) = \frac{1}{K} \sum_{n=q}^{q+K-1} |x(n) - x(n+k)|, \quad k = 0, 1, \dots, N. \tag{1}$$

Let q be the beginning sampling point of the pitch extracting speech part. N is the number of sampling points used to find the waveform period. The minimum of $a(k)$ is zero in case the input voice signal $x(n)$ is exactly periodic. However, because voice speech is a quasi-periodic signal, the AMDF will seldom fall to zero but will only fall to a very low value [14]. After obtaining the waveform period K , the time period T_0 can be computed using K and a speech sampling frequency (F_s) as follows.

$$T_0 = \frac{K}{F_s} \tag{2}$$

The sampling frequency used in this work is 11,025 Hz. Then the fundamental frequency (F_0) can be calculated using the next equation.

$$F_0 = \frac{1}{T_0} \tag{3}$$

In this work, the fundamental frequencies are computed using the Snack Sound Toolkit [15]. Since sometimes, there are unvoiced parts at the beginning and the ending of a syllable, after obtaining the fundamental frequencies, the only longest consecutive part of pitch contour (fundamental frequencies) without a zero value is used. The steps of pitch frequency extraction are shown in Fig. 4.

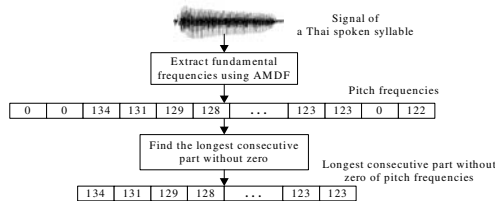


Fig. 4. Steps of pitch frequency extraction

After that the pitch frequencies are upsampled or downsampled to fit the number of neural network inputs as explained in the next section.

3.2 Pitch Feature Representation for a Neural Network

Since the numbers of pitch frequencies obtained from syllables are not equal, depending on the duration of a spoken syllable. Before training and classifying genders, the pitch frequencies of each Thai syllable are upsampled or downsampled to fit the number of neural network inputs, using the steps shown in Fig. 5.

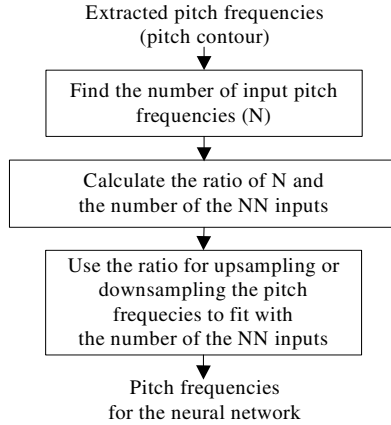


Fig. 5. Pitch frequency conversion for a neural network

Firstly, the number of pitch frequencies extracted from a spoken syllable is determined. Then the ratio of the number of pitch frequencies and the neural network inputs is calculated and used for upsampling or downsampling pitch frequencies in order to obtain the number of pitch frequencies that equals the number of neural network inputs. Finally, the pitch frequencies are normalized into the range of 0 and 1 by dividing the pitch frequencies with 400, which is the highest pitch frequency set in the pitch extraction. The obtained normalized pitch frequencies are used as the inputs for the neural network classifier.

3.3 Gender Classification by a Neural Network Classifier

The multi-layer perceptron neural network (MLP) consisting of three layers of neurons, which are the input layer, the hidden layer, and the output layer, is used as a gender classifier. The neural network is trained by the Backpropagation algorithm. The considered transfer functions in the hidden layer are hyperbolic tangent sigmoid (Tansig), radial basis (Radbas), and log sigmoid (Logsig). In the output layer, the Logsig transfer function is used. The extracted pitch frequencies obtained from all five Thai tone syllables are used to train the neural network. To identify the gender of a speaker, the pitch frequencies computed from a Thai spoken syllable are used as the input for the neural network. The classification result is determined from the output of

the female node and the male node, as shown in Fig. 2. If the output of the female node has a higher value than the output of the male node, the gender is identified as a female. On the other hand, if the output of male node provides a higher value than the output of the female node, the gender is identified as a male.

4 Experimental Results

In the experiments, the training and testing speech are recorded in the 16-bit PCM format at 11,025 samples per second. The VXi TalkPro Max headset and Andrea USB audio adapter are used to record speech signals. The training set consists of 20 females and 20 males. The syllables with five Thai tones are used in training. There are 2 test sets. The first consists of 20 different females and 20 different males speaking the same syllables used in training. The second test set comprises 70 females and 60 males speaking different syllables from those used in training. The experiments are divided into 3 parts: the gender classification using a syllable, the gender classification using different parts of a syllable, and gender classification using a different syllable from training.

4.1 Gender Classification Using a Syllable

The gender classification using the whole Thai syllable with the neural network is studied in this part. The MLP neural network consisting of 40 input nodes, 50 hidden nodes and 2 output nodes is used. In training, the pitch frequencies of all five different Thai tone syllables from 40 speakers, consisting of 20 females and 20 males, are combined and used to create a single neural network. In testing, the five-tone syllables from another set of 20 females and 20 males are used. The classification results are shown in Table 1.

Table 1. Gender classification results using a Thai syllable categorized by Thai tones

Classifier \ Tones	NN & Hyperbolic Tangent Sigmoid (Tansig)	NN & Radial Basis (Radbas)	NN & Log Sigmoid (Logsig)	HMM with MFCC
Mid	100%	100%	100%	97.5%
Low	100%	100%	100%	97.5%
Falling	95%	95%	95%	92.5%
High	100%	100%	97.5%	100%
Rising	95%	97.5%	95%	97.5%
Average	98%	98.5%	97.5%	97%

The result reveals that nearly the same average gender classification rates of 98%, 98.5%, and 97.5% are obtained from applying the different transfer functions: hyperbolic tangent sigmoid, radial basis, and log sigmoid transfer functions, respectively. We have compared these results with those from the method using Mel Frequency Cepstral Coefficient (MFCC) with a Hidden Markov Model classifier (HMM). Our method using pitch frequency contour with the NN classifier provides the better

results, in particular, by using the radial basis transfer function. There are 37 from 40 speakers, which is 92.5% of speakers, that their genders are accurately recognized from any tones of five Thai tone syllables. The detail results of each speaker are shown in Table 2.

Table 2. Gender classification results of 40 test speakers grouped by Thai tones

Speaker No. / Tones	1-7	8	9-18	19	20-21	22	23-27	28	29-36	37	38-40
Mid											
Low											
Falling						X				X	
High											
Rising				X							

Remark: X means incorrect gender classification

The results reveal that any tones can be used to identify the gender with the high accuracy rates.

4.2 Gender Classification Using Different Parts of a Syllable

Gender classification using different parts of a syllable is investigated in this part. The pitch frequencies extracted from the first 50%, the middle 50%, and the last 50% of a syllable are used. The neural networks with 20 input nodes, 25 hidden nodes and 2 output nodes are applied and the details of gender classification results are shown in Table 3.

Table 3. Gender classification results using parts of a syllable

Parts / Tones	First Half Part of a Syllable			Middle Half Part of a Syllable			Last Half Part of a Syllable		
	Tan sig	Rad bas	Log sig	Tan sig	Rad bas	Log sig	Tan sig	Rad bas	Log sig
Mid	100%	100%	100%	100%	100%	100%	100%	100%	100%
Low	97.5%	100%	100%	97.5%	100%	100%	97.5%	97.5%	97.5%
Falling	97.5%	97.5%	97.5%	90%	92.5%	92.5%	92.5%	92.5%	92.5%
High	100%	100%	100%	97.5%	95%	97.5%	95%	95%	95%
Rising	100%	100%	100%	92.5%	92.5%	92.5%	92.5%	92.5%	92.5%
Avg	99%	99.5%	99.5%	95.5%	96%	96.5%	95.5%	95.5%	95.5%

Interestingly, only half of a syllable can be used to identify the gender of a speaker quite accurately. The first half part of Thai syllables outperforms the middle and the last half part of the syllables. When comparing among three transfer functions, the radbas transfer function performs better than others. The average correct gender classification rates of 99.5%, 96% and 95.5% are attained from the first, middle and last half of syllables, respectively.

4.3 Gender Classification Using a Different Syllable from Training

In this experiment, syllables and speakers used in gender classification are different from those used in training. The same training set as used in the aforementioned experiments is used in training and there are 130 speakers consisting of 70 females and 60 males used in testing. Each test speaker enunciates 5 syllables having different tones. When the neural network with the radial basis transfer function is applied, the gender classification results are obtained as shown in Table 4.

Table 4. Gender classification results when using a different syllable from training

Tones \ Syllable Parts	Whole Syllable	First Half Part of a Syllable
Mid	97.7%	99.2%
Low	99.2%	99.2%
Falling	96.9%	95.4%
High	99.2%	98.5%
Rising	96.9%	99.2%
Average	98%	98.3%

The results show that when using the whole syllable to determine a gender, the classification rates of 97.7%, 99.2%, 96.9%, 99.2%, and 96.9% are obtained from mid, low, falling, high, and rising tone syllables, respectively. The gender classification rates of 99.2%, 99.2%, 95.4%, 98.5%, and 99.2% can be achieved from the first half part of mid, low, falling, high, rising tone syllables, respectively. On average, the first half of a Thai syllable slightly outperforms the whole syllable (98.3% vs. 98%). When comparing the gender classification using the different Thai syllables from training with the gender classification using the same Thai syllables in training, the comparable classification rates of 98% vs. 98.5% for the whole syllable and 98.3% vs. 99.5% for the first half part of a syllable can be obtained. The results indicate the good generalization performance of using pitch contour of a Thai syllable and the neural network for the gender identification task.

5 Conclusions

This work proposed the method to identify a gender based on a Thai syllable using the AMDF and the MLP neural network. The experimental results show that Thai syllables with any tones can be used for gender classification through pitch contour. By using the whole syllable, the average correct gender classification rate up to 98.5% can be achieved from the MLP neural network with the radial basis transfer function in the hidden layer. Furthermore by using a part of a syllable, we found that the first half of a syllable is sufficient for gender classification and it can give the slightly higher accuracy than the whole syllable, 99.5% vs. 98.5%. Moreover, even using the whole syllable and the first half part of a syllable that is different from those used in training,

the high average gender classification rates of 98% and 98.3% can be achieved, respectively. The results show that our proposed method using the pitch contour of the first half of a Thai syllable or a whole Thai syllable from any tones with the MLP neural network classifier is an efficient method for gender identification.

References

1. Benzeghiba, M., Mori, R.D., et al.: Impact of Variabilities on Speech Recognition. In: SPECOM, 11th International Conference Speech and Computer (2006)
2. Benzeghiba, M., Mori, R.D., et al.: Automatic Speech Recognition and Speech Variability: a Review. *Speech Communication* 49, 763–786 (2007)
3. Gharavian, D., Ahadi, S.M.: Statistical Evaluation of the Effect of Gender on Prosodic Parameters and Their Influence on Gender-dependent Speech Recognition. In: ICICS (2007)
4. Kotti, M., Kotropoulos, C.: Gender Classification in Two Emotional Speech Databases. In: Proc. ICPR, 19th International Conference on Pattern Recognition (2008)
5. Sigmund, M.: Gender Distinction Using Short Segments of Speech Signal. *International Journal of Computer Science and Network Security* 8(10) (2008)
6. Kim, H.J., Bae, K., Yoon, H.S.: Age and Gender Classification for A Home-Robot Service. In: 16th IEEE International Conference on Robot & Human Interactive Communication, pp. 122–126 (2007)
7. Tangwongsan, S., Phoophuangpairoj, R.: Boosting Thai Syllable Speech Recognition Using Acoustic Models Combination. In: ICCEE, pp. 568–572 (2008)
8. Meng, Z., Chen, Y., Li, X.: Fundamental Frequency Survey of Mandarin Monophthongs. In: Proc. WESPAC, 9th Western Pacific Acoustic Conference (2006)
9. Azghadi, S.M.R., Bonyadi, M.R., Sliahhosseini, H.: Gender Classification Based on Feed-Forward Backpropagation Neural Network. In: IFIP International Federation for Information Processing, vol. 247, pp. 299–304 (2007)
10. Ting, H., Yingchun, Y., Zhaohui, W.: Combining MFCC and Pitch to Enhance the Performance of the Gender Recognition. In: ICSP (2006)
11. Tangwongsan, S., Po-Aramsri, P., Phoophuangpairoj, R.: Highly Efficient and Effective Techniques for Thai Syllable Speech Recognition. In: Maher, M.J. (ed.) ASIAN 2004. LNCS, vol. 3321, pp. 259–270. Springer, Heidelberg (2004)
12. Tunthangthum, A.: Tone Recognition for Thai. In: IEEE Asia-Pacific Conference on Circuits and Systems, APCCAS, pp. 157–160 (1998)
13. Rabiner, L.R., Schafer, R.W.: Introduction to Digital Speech Processing. *Foundations and Trends in Signal Processing* 1, 1–194 (2007)
14. Vishnubhotla, S., Espy-Wilson, C.: An Algorithm for Multi-Pitch Tracking in Co-Channel Speech. In: ICSLP (2008)
15. Sjölander, K.: The Snack Sound Toolkit, Department of Speech, Music and Hearing, KTH Royal Institute of Technology, <http://www.speech.kth.se/snack/>

Gender Classification Based on Support Vector Machine with Automatic Confidence

Zheng Ji¹ and Bao-Liang Lu^{1,2,*}

¹Center for Brain-Like Computing and Machine Intelligence
Department of Computer Science and Engineering

²MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems
Shanghai Jiao Tong University
800 Dong Chuan Rd., Shanghai 200240, China
{jizheng, bllu}@sjtu.edu.cn

Abstract. In this paper, we propose a support vector machine with automatic confidence (SVMAC) for gender classification based on facial images. Namely, we explore how to incorporate confidence values introduced in each primitive training sample and compute these values automatically into machine learning. In the proposed SVMAC, we use both the labels of training samples and the label confidence which projected by a monotonic function as input. The main contribution of SVMAC is that the confidence value of each training sample is calculated by some common algorithms, such as SVM, Neural Network and so on, for gender classification. Experimental results demonstrate that SVMAC can improve classification accuracy dramatically.

Keywords: Support Vector Machine, Feature Extraction, Gender Classification.

1 Introduction

Due to its wide application in human-computer interaction, such as vital statistics, identity appraisal, visual surveillance and robot vision [1], gender classification based on facial images has attracted many researchers' attention.

One of the most challenging problems is to devise a proper classification algorithm to classify gender information of faces. In other words, we need to select a better classifier to improve the classification performance. Among all kinds of recognition algorithms [2] [3] [4] [5], support vector machine (SVM) is one of the most popular classification methods, providing a sound theoretic basis for constructing classification models with high generalization ability. Li and Lu [6] brought forward a framework based on multi-view gender classification where a trained layer support vector machine (LSVM) is utilized to recognize the angle of each facial image and classify its gender. To develop new machine learning algorithms and improve the performance of existing machine learning methods, it is very important for us to consider the problem how SVM can be ameliorated. Ji *et al* [7] proposed a support vector machine with confidence (SVMC) labeled manually. But before training, the confidence of each training sample

* Corresponding author.

must be labeled manually. Thus, when the number of training samples is very large we must spend much time in labeling the confidence. Furthermore, we can not guarantee all these labeled confidence values are reasonable. To explore how to label rational confidence for each training sample automatically, we propose the support vector machine with automatic confidence (SVMAC).

The remaining part of the chapter is organized as follows: Section 2 described the proposed SVMAC model in detail. Experiments are presented in Section 3. Some conclusions and discussions on future work are outlined in Section 4.

2 The Proposed SVMAC Model

The quadratic programming problems for the standard and soft margin forms of traditional SVM [8] [9] [10] [7] can be expressed as

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

and

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + D \sum_i \xi_i^2 \\ \text{s.t.} \quad & \forall i, y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0 \end{aligned} \tag{2}$$

respectively. One way of incorporating confidence values is to re-scale the soft margin as follows,

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + D \sum_i \xi_i^2 \\ \text{s.t.} \quad & \forall i, y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq t(\pi_i) - \xi_i, \\ & \xi_i \geq 0 \end{aligned} \tag{3}$$

where $t(\pi_i)$ is a monotonic function to scale the confidence, namely

$$t(\pi_i) = h \cdot \pi_i, \quad \frac{1}{2} \leq \pi_i < 1 \tag{4}$$

where h is the scale parameter.

Existing work reported that the support vectors obtained by a support vector machine tend to be those training samples that people can not discriminate well [2] [11]. Based on this fact, we propose a modified support vector machine. First, we divide all the training samples into two disjointed subsets \mathcal{U} and \mathcal{V} ($\mathcal{X} = \mathcal{U} \cup \mathcal{V}$), which are later treated in a different way in the training algorithm. Then, we put the training samples in \mathcal{U} with confidence π_i less than 1, and the remaining training samples in \mathcal{V} with confidence π_i equal to 1. In essence, \mathcal{U} contains the training samples that tend to be support vectors after training. In the following, we denote the number of training samples in \mathcal{U} and \mathcal{V} by n_u and n_v , respectively.

According to Eq. (3) for training subset \mathcal{U} and Eq. (1) for training subset \mathcal{V} , we can express the quadratic programming problem for soft margin form as follows:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + D \sum_{i=1}^{n_u} \sigma_i^2 + C \sum_{i=1}^{n_v} \xi_i, \\ \text{s.t.} \quad & \forall 1 \leq i \leq n_u, \\ & y_i^u (\mathbf{w}^T \mathbf{u}_i + b) = t(\pi_i) - \sigma_i, \\ & \forall 1 \leq i \leq n_v, \\ & y_i^v (\mathbf{w}^T \mathbf{v}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0. \end{aligned} \tag{5}$$

Using the standard Lagrangian dual technique, we get the following dual form:

$$\begin{aligned} \min_{\lambda, \alpha} \quad & \frac{1}{2} \left(\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{i=1}^{n_v} \alpha_i y_i^v \mathbf{v}_i \right)^T \\ & \left(\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{i=1}^{n_v} \alpha_i y_i^v \mathbf{v}_i \right) \\ & - \sum_{i=1}^{n_u} t(\pi_i) \lambda_i + \frac{1}{4D} \sum_{i=1}^{n_u} \lambda_i^2 - \sum_{i=1}^{n_v} \alpha_i \\ \text{s.t.} \quad & \forall 1 \leq i \leq n_u, \quad 0 \leq \lambda_i < +\infty, \\ & \forall 1 \leq i \leq n_v, \quad 0 \leq \alpha_i \leq C, \\ & \sum_{i=1}^{n_u} \lambda_i y_i^u + \sum_{i=1}^{n_v} \alpha_i y_i^v = 0. \end{aligned} \tag{6}$$

However, it is a considering problem how to label the confidence of training samples reasonably. In SVMC [7] the confidence of all the training samples is labeled manually. But if the number of the training samples is large, we must spend much time in labeling their confidence values. Moreover, we can not guarantee that all the labeled confidence values are reasonable because people’s action on making them is very subjective. Therefore, we suggest using logical methods to divide the training samples into the two sets \mathcal{U} and \mathcal{V} . The algorithm ALC of labeling the confidence is displayed in Algorithm 1.

As a matter of fact, the size of the distance between a training sample and decision boundary γ suggests whether the sample can be discriminated well. Obviously, the training sample which is far from the decision boundary can tend to be discriminated and should be appended into \mathcal{V} . Otherwise, it need to be added in \mathcal{U} . Therefore, the automatic mark is coincident with the manual label on the confidence of training samples.

Now we take into account the performance of SVMAC in some situation. According to the algorithm ALC and SVMAC defined in Eq. (5) that makes use of the method of probability statistics [7] [12] [13], we set the confidence values in \mathcal{U} less than 1. Those samples marked by small circles (green) are shown in Fig. 1 and the right figure of Fig. 2. We can observe that the decision boundary is changed if the confidence values of the support vectors in \mathcal{U} are assigned by employing the algorithm ALC where the sample set $\Gamma = \{(x_i, y_i) | 1 \leq i \leq N\}$ is trained by SVM. We can conclude that the movement of the decision boundary is identical to the one in SVMC [7].

Algorithm 1. ALC

Step 1: The sample set $\Gamma = \{(x_i, y_i) | 1 \leq i \leq N\}$ is trained by utilizing some algorithms such as SVM, Adaboost, Neural Network, and so on. Thus, a decision hyperplane γ , namely, a classifier can be obtained;

Step 2: Computing these distances between all the samples in Γ and the hyperplane γ and Obtaining the distance set $\Omega = \{d_i | \text{the distance between the } i\text{-th sample and } \gamma\}$;

Step 3: Given threshold value σ ,
for all i from 1 to N
 if $d_i < \sigma$,
 the sample (x_i, y_i) is added in \mathcal{U} ;
 else
 (x_i, y_i) is added in \mathcal{V} ;
 end if
end for

Step 4: The confidence values of the samples in \mathcal{V} are set to 1.0 while the ones in \mathcal{U} is projected onto the confidence space $[\frac{1}{2}, 1)$ according to the distances.

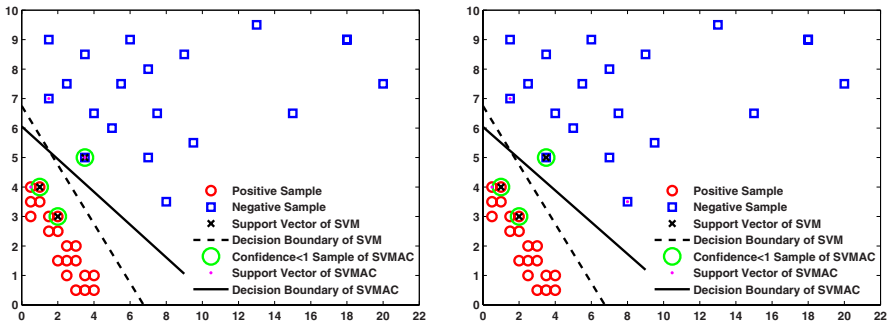


Fig. 1. Illustration of movement of decision boundary caused by the proposed SVMAC. The scale parameter h for SVMAC in Eq. (4) is set to 1.0 and 0.1 in the left and right figures, respectively.

We regard the support vectors obtained by means of SVM as the training samples close to noise. Therefore, we should assign them with confidence values less than 1. By training the proposed SVMAC on all the training samples with proper confidence values, we obtain the decision boundaries as shown in Fig. 1. From this figure, we can see that if the support vectors obtained by the traditional SVM are assigned with appropriate confidence values, some of them may be turned into non-support vectors after applying SVMAC. The decision boundary obtained by SVMAC can be regarded as a fitting achieved by training the samples in which some noise is removed. Therefore, the decision boundary obtained by SVMAC is superior to that obtained by traditional SVM. For example, the lower left training samples in Fig. 1 are much denser and closer to the boundary than the upper right training samples, the movement of separation boundary from the lower left corner to the upper right corner caused by the proposed SVMAC no doubt yields a better separation than that of traditional SVM.

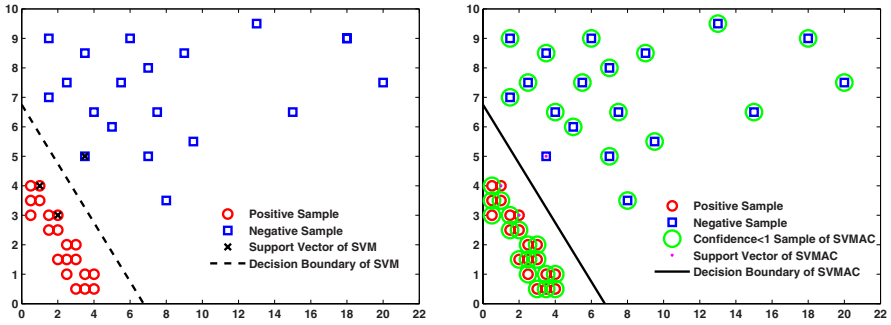


Fig. 2. Illustration of the decision boundaries formed by SVM (left) and SVMAC (right), where we only assign the confidence values (less than 1) to non-support vectors in \mathcal{V} for training SVMAC and do not consider any confidence values for support vectors in \mathcal{U}

Table 1. Description of training and test data based on facial images

Data Set Description		Total	Male	Female	Training	Test
CAS-PEAL	PD00	1040	595	445	311*2	418
	PD15	939	516	423	296*2	347
	PD30	939	516	423	296*2	347
	PM00	1039	595	444	310*2	419
	PM15	938	516	422	295*2	348
	PM30	938	516	422	295*2	348
	PU00	1040	595	445	311*2	418
	PU15	939	516	423	296*2	347
	PU30	939	516	423	296*2	347
	TOTAL	8751	4881	3870	5412	3339
FERET	PM00	992	589	403	282*2	428
BCMI	PM00	1045	529	516	361*2	323
TOTAL	TOTAL	10788	5999	4789	6698	4090

From the angle of the confidence, the decision boundaries in Fig. 1 are the most superior boundaries. However, the decision boundaries produced by traditional SVM and the proposed SVMAC are the same as shown in Fig. 2, where only the non-support vectors in \mathcal{V} are assigned with confidence values less than 1 and none of support vectors in \mathcal{U} is assigned with confidence value. From this figure, we can see that the confidence values less than 1 assigned to non-support vectors in \mathcal{V} don't affect the decision boundary. In other words, after some non-support vectors in SVM are marked by the confidence values less than 1 through using ALC, the whole classification accuracy will not be decreased.

3 Experiments

To evaluate the performance of SVMAC, we select the gender classification problem based on multi-view facial images in the CAS-PEAL face database [14] and frontal

Table 2. Description of the mean accuracy denoted by *MA* (%) and the standard deviation defined as *SD* (%) caused by SVM and SVMAC with RBF kernel used by the first row and the second row in each group in turn on $F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8$ corresponding to MLGBP-CCL, MLGBP-LDA, LGBP-CCL, LGBP-LDA, MLBP, LBP, Gabor, and gray respectively, where CAS-PEAL, FERET and BCMI are represented by C, F, and B in front of these description names, and m (the number of window blocks) is ranged from 5×5 to 10×10

Description	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
	<i>MA/SD</i>	<i>MA/SD</i>	<i>MA/SD</i>	<i>MA/SD</i>	<i>MA/SD</i>	<i>MA/SD</i>	<i>MA/SD</i>	<i>MA/SD</i>
C-PD00	99.1/0.4	99.9/0.2	96.7/11.4	99.6/0.4	95.5/2.9	94.2/10.2	90.6/16.3	92.5/4.9
	99.3/0.3	100.0/0.0	97.4/4.8	99.9/0.2	96.4/1.1	95.3/4.4	92.6/15.1	93.8/4.3
C-PD15	99.3/1.1	100.0/0.0	97.4/7.5	99.7/0.1	94.9/1.1	93.5/3.0	88.7/26.0	91.3/1.1
	99.4/0.9	100.0/0.0	97.7/7.3	99.8/0.1	95.6/0.9	94.6/2.1	90.5/17.2	92.6/2.0
C-PD30	98.9/0.4	99.9/0.2	96.4/6.2	99.8/0.4	92.7/2.0	92.2/3.2	89.9/13.1	89.4/1.5
	99.1/0.6	100.0/0.0	96.8/4.8	100.0/0.0	93.3/2.1	93.2/1.9	90.4/8.4	90.6/2.5
C-PM00	99.6/0.2	100.0/0.0	97.3/10.6	100.0/0.0	96.1/2.7	94.5/10.4	91.6/26.3	94.6/6.0
	99.6/0.3	100.0/0.0	97.5/10.8	100.0/0.0	96.6/3.3	95.5/8.5	92.7/16.7	95.4/3.6
C-PM15	99.6/0.2	99.9/0.1	97.1/2.9	99.7/0.3	95.1/0.9	94.2/14.6	92.7/30.2	94.5/7.4
	99.7/0.8	100.0/0.0	97.3/2.7	99.9/0.2	95.7/2.5	94.7/15.3	93.2/23.4	95.0/6.9
C-PM30	98.9/0.3	99.9/0.1	96.3/6.8	99.7/0.5	93.7/9.1	92.4/15.3	91.9/17.4	92.1/1.5
	99.4/0.2	100.0/0.0	96.9/2.8	100.0/0.0	95.1/6.0	93.2/11.3	93.1/8.3	92.6/2.0
C-PU00	99.4/0.5	100.0/0.0	96.7/5.4	99.9/0.1	95.4/4.2	94.5/4.7	90.2/33.2	89.0/28.4
	99.6/0.5	100.0/0.0	97.6/5.6	100.0/0.0	96.3/2.4	95.3/4.8	91.6/18.4	92.0/17.9
C-PU15	99.3/0.2	100.0/0.0	97.6/1.3	99.9/0.1	96.0/4.2	95.6/1.8	92.6/8.9	89.3/16.5
	99.9/0.2	100.0/0.0	98.3/2.5	100.0/0.0	96.6/0.8	96.0/2.5	93.1/6.5	90.2/18.8
C-PU30	98.8/1.4	100.0/0.0	96.9/8.6	99.9/0.2	94.0/4.0	92.8/5.1	89.0/21.1	88.2/9.1
	99.2/0.5	100.0/0.0	97.0/6.3	100.0/0.0	95.0/1.2	93.6/5.5	90.2/13.0	89.6/5.9
F-PM00	96.8/2.3	99.6/0.2	94.6/11.6	98.8/2.3	93.7/1.3	92.3/9.0	90.4/9.8	89.5/3.7
	97.2/2.4	99.7/0.2	95.1/8.8	99.1/0.6	93.8/1.0	93.1/3.9	91.6/15.1	91.2/4.6
B-PM00	98.8/1.2	100.0/0.0	97.3/8.3	99.4/1.4	96.3/2.0	96.2/3.2	93.4/10.2	95.4/2.1
	99.0/0.7	100.0/0.0	98.1/0.7	99.7/0.2	97.2/0.9	97.3/0.9	95.1/6.1	95.7/1.2

face pictures in FERET¹ and BCMI² databases as a benchmark problem, respectively, and make some comparative studies. The total 10788 different-pose facial images are organized into 11 groups in each of which the numbers of training and test samples are 70% and 30% of the whole group, respectively (See Table 1).

In this paper, we use gray, Gabor, local binary pattern (LBP) [15] [16], multi-resolution local binary pattern (MLBP) [17], local Gabor binary pattern (LGBP) [18], and multi-resolution local Gabor binary pattern (MLGBP) approaches to extract the features of each facial image. Thereinto, the MLGBP feature as input of SVM [8] [9] [10] and SVMAC classifiers is derived by combining multi-resolution analysis, Gabor characteristic and uniform LBP histograms. All experiments were performed on a Pentium fourfold CPU (2.83GHz) PC with 8GB RAM.

¹ <http://www.frvt.org/FERET/default.htm>

² BCMI face database is set up and packed up by the Center for Brain-Like Computing and Machine Intelligence Shanghai Jiao Tong University, Shanghai, China.

From Table 2 we conclude that the average classification accuracy caused by SVMAC is higher than SVM on the same face feature, and the standard deviation of classification precision brought by SVMAC is lower than SVM. What's the more, the performance improvement of SVMAC is obvious for Gray, Gabor, LBP and MLBP features and the maximum improvement accuracy obtained between SVMAC and SVM reaches 3.0%. But for MLGBP-CCL, LGBP-CCL, MLGBP-LDA and LGBP-LDA features, because all the accuracies are very high, SVMAC improve the classification performance a little only. These indicate that SVMAC improves the classification performance compared to traditional SVM, where the parameter C in Eqs. (1) and (6) is consistent. In addition, we observe that the classification performance is also dependent on the distributions of training samples. Generally speaking, there are two kinds of sample distributes. One is dense and the other is sparse, such as in Fig. 1. In this situation, if the confidence values less than 1 are set for the support vector samples, the decision boundary obtained by SVMAC will favor the sparse samples in comparison with traditional SVM. Consequently, from the experimental results and theoretical analysis, by modifying the confidence values of the support vector samples, we can separate the data samples more reasonably.

4 Conclusions and Future Work

We have proposed a novel support vector machine with automatic confidence, i.e., SVMAC. The most important advantage of this presented SVMAC over traditional SVM is that some explicit human prior knowledge estimated by the algorithm ALC on training samples can be easily incorporated into learning. We have derived the quadratic programming problem for SVMAC and analyzed its performance theoretically. Experimental results on a gender classification problem based on facial images indicate that this proposed method can improve classification accuracy dramatically. As future work, we would like to give a bound for the improvement on classification accuracy about SVMAC and apply it to other real-world pattern classification problems, such as text classification, age estimation and object recognition.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Grant No. 60773090 and Grant No. 90820018), the National Basic Research Program of China (Grant No. 2009CB320901), and the National High-Tech Research Program of China (Grant No. 2008AA02Z315).

References

1. Schmitt, L.A., Gruver, W.A., Ansari, A.: A robot vision system based on two-dimensional object-oriented models. *IEEE Transactions on Systems, Man and Cybernetics* 16(4), 582–589 (1986)
2. Graf, A., Wichmann, F., Bühlhoff, H., Schölkopf, B.: Classification of faces in man and machine. *Neural Computation* 18(1), 143–165 (2005)

3. Doumpos, M., Zopounidis, C., Golfinopoulou, V.: Additive support vector machines for pattern classification. *IEEE Transactions on System, Man, and Cybernetics* 37(3), 540–550 (2007)
4. Peng, J., Heisterkamp, D.R., Dai, H.K.: Lda/svm driven nearest neighbor classification. *IEEE Transactions on Neural Networks* 14(4), 158–163 (2003)
5. Chen, J.H., Chen, C.S.: Reducing svm classification time using multiple mirror classifiers. *IEEE Transactions on System, Man, and Cybernetics, Part B* 34(2), 1173–1183 (2004)
6. Li, J., Lu, B.L.: A framework for multi-view gender classification. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) *ICONIP 2007, Part I. LNCS*, vol. 4984, pp. 973–982. Springer, Heidelberg (2007)
7. Ji, Z., Yang, W.Y., Wu, S., Lu, B.L.: Encoding human knowledge for visual pattern recognition. In: *The 5th International Symposium on Neural Networks* (2008)
8. Cristianini, N., Shawe-Taylor, J.: *An introduction to support vector machines and other kernel-based learning methods*. Publishing House of Electronics Industry (2004)
9. Sloin, A., Burshtein, D.: Support vector machine training for improved hidden markov modeling. *IEEE Transactions on Signal Processing* 56(1), 172–188 (2008)
10. Williams, P., Li, S., Feng, J., Wu, S.: A geometrical method to improve performance of the support vector machine. *IEEE Transactions on Neural Networks* 18(3), 942–947 (2007)
11. Feng, J., Williams, P.: The generalization error of the symmetric and scaled support vector machine. *IEEE Transactions on Neural Networks* 12(5), 1255–1260 (2001)
12. Osuna, E., Freund, R., Girosi, F.: An improved training algorithm for support vector machines. *Neural Networks for Signal Processing*, 276–285 (1997)
13. Vapnik, V.N.: *Statistical learning theory*. Wiley, New York (1998)
14. Gao, W., Cao, B., Shan, S.G., et al.: The cas-peal large-scale chinese face database and baseline evaluations. Technical report of JDL (2004), http://www.jdl.ac.cn/~peal/peal_tr.pdf
15. Ojala, T., Pietikainen, M., Mäeopaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE transactions on Pattern Analysis and Machine Intelligence* 24(7), 971–987 (2002)
16. Ahonen, T., Hadid, A., Pietikainen, M.: Face recognition with local binary patterns. In: Pajdla, T., Matas, J(G.) (eds.) *ECCV 2004. LNCS*, vol. 3021, pp. 469–481. Springer, Heidelberg (2004)
17. Lian, H.C., Lu, B.L.: Multi-view gender classification using multi-resolution local binary patterns and support vector machines. *International Journal of Neural Systems* 17(6), 479–487 (2007)
18. Xia, B., Sun, H., Lu, B.L.: Multi-view gender classification based on local gabor binary mapping pattern and support vector machines. *IEEE International Joint Conference on Neural Networks*, 3388–3395 (2008)

Multiple Occluded Face Detection Based on Binocular Saliency Map

Bumhwi Kim¹, Sang-Woo Ban², and Minho Lee^{3,*}

¹ Dept. of Sensor and Display Engineering, Kyungpook National University
1370 Sankyuk-Dong, Puk-Gu, Taegu 702-701, Korea

² Dept. of Information and Communication Engineering, Dongguk University 707
Seokjang-Dong, Gyeongju, Gyeongbuk, 780-714, Korea

³ School of Electrical Engineering and Computer Science, Kyungpook National University
1370 Sankyuk-Dong, Puk-Gu, Taegu 702-701, Korea

bhkim@ee.knu.ac.kr, swban@dongguk.ac.kr, mhlee@knu.ac.kr

Abstract. In this paper, we propose a novel occluded face detection model which can detect multiple occlusions in a stereo image. The biologically motivated face preferable selective attention model localizes candidate regions for human faces in a natural scene, and then the Adaboost based face and eye detection process works for those localized candidate areas to check whether the areas contain a human face. In order to detect each facial area in multiple occluded faces, we use depth information of an eye, which is obtained by a binocular saliency map model. If the facial local features are similar depth information, we use the conventional Adaboost algorithm to localize the face area, and mask off the facial region. Then, we check the next area in a distance point of view whether it has an occlusion by Auto-associative multilayer perceptron (AAMLN) at 3 divided candidate regions. Finally, we implement an efficient model which can detect not only faces in a stereo image but also multiple occluded facial regions in an input scene. Experimental results show that the proposed model successfully localizes multiple occluded faces.

Keywords: Multiple occlusion, Occlusion detection, Binocular saliency, saliency map, Face detection, AAMLN, AdaBoost.

1 Introduction

In the last decades, face detection is one of hottest issues as well as face recognition. Face and facial expression recognition have attracted much attention though they have been studied for more than 20 years by psychophysicists, neuroscientists, and engineers [1]. Numerous methods have been developed to localize or detect faces in a visual scene [1, 2]. M. Yang et al, [1] have reviewed and classified those face detection methods into four major categories such as the knowledge-based methods, the feature invariant approaches, the template matching methods, appearance-based methods. The-state-of-the-art of the face detection shows excellent performance [3].

* Corresponding author.

But, the performance was still limited under constrained environment. There exist various environmental components to deteriorate face detection performance, such as shadow, occlusion, head rotation, view point change, various illuminations and so on. Many researchers are trying to develop more robust face detection and recognition methods, but no specific method has yet shown comparable performance with a human being.

Biologically inspired vision system may provide a critical clue to overcome the limitations of the current artificial vision system. Recently, biologically motivated approaches have been developed by L. Itti., T. Poggio, and C. Koch [4, 5, 6]. And, attention models were introduced for face detection [7, 8]. However, they have not shown plausible results for the face attention problem in complex scenes including multiple occlusion until now. Conventional face detection models based on an AdaBoost algorithm show good performance in real time environment even if they are not perfectly working [9].

In this paper, we propose a new face detection model to localize face areas even when multiple faces are occluded. The conventional methods based on the AdaBoost algorithm can work well for single face detection, but it doesn't work for detecting a multiple occluded faces because the AdaBoost algorithm uses a Haar-like feature for complete facial features such as a distance of intensity difference between two eyes, cheeks, nose and mouth. However, there needs to detect a face with partial information such as crowded area for airport and/or downtown street, in which some of facial features are not available caused by occlusion. In order to detect facial areas robustly with and without complete facial features, we propose a new method based on biologically motivated multiple occluded face detection. We localize the candidate area to search a facial feature by a face preferable selective attention model, and apply the wavelet based Adaboost algorithm to find a facial feature such as eye region. Then, we consider depth information of the facial features to sequentially identify frontal and behind facial features in multiple occluded face images, in which a biologically inspired binocular saliency map model is used for obtaining depth information [9]. We finally localize the suitable size of a facial region based on the 3D facial feature information. Also, we use an auto-associative multilayer perceptron (AAMLN) model to identify the occluded local region. Finally, we implement a face detection model which can detect not only faces in stereo image but also multiple occluded facial regions. Experimental results show that the proposed model can successfully detect an each facial region from multiple occluded faces.

This paper is organized as follows; Section 2 describes the proposed model including a face preferable selective attention, a binocular saliency for depth information of a local facial feature, and AAMLN for detecting the occluded region. Experimental results will be followed in Section 3. At last, further works with conclusion are discussed in Section 4.

2 Proposed Model

Fig. 1 shows the overall architecture of the proposed model. The selective attention model considers a face color preferable intensity, an RG color opponent and edge information for reflecting human preference for faces. Thus, the proposed selective

attention model generates a saliency map (SM) for an input scene, which pop-outs face candidate areas having the face-like low level features. The face preferable SM model reduces region of interests (ROI) in an input scene, which plays important role for decreasing processing time of face detection [10]. Moreover, in order to reject non-face areas and correctly localize face areas in the selected face candidate areas, we consider a well-known AdaBoost algorithm based on Haar-like form features. Using binocular face preferable SM model, we can obtain distance information to a local feature area in a face region. The AAMLP uses to detect whether a candidate facial region is occluded or not.

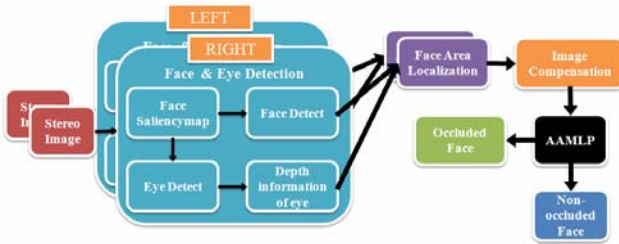


Fig. 1. The proposed model flow chart

2.1 Face Preferable Selective Attention with Adaboost

In order to implement a human-like visual attention function, we consider the simplified bottom-up SM model proposed in [11]. The SM model reflects the functions of the retina cells, the lateral geniculate nucleus (LGN) and the visual cortex. In order to provide the proposed model with face color preference property, the skin color preferable intensity feature is considered together with the original intensity feature. According to a given task to be conducted, those two intensity features are differently biased. For face preferable attention, a skin color preferable intensity feature works for a dominant feature in generating an intensity feature map. And the real color components R, G, B, Y are extracted using normalized color coding [11]. According to our experiments, the real color component R among 4 real color components shows dominant contribution for face color plausible filtering. Moreover, RG color opponent coding features also show a discriminate characteristic between face and non-face area. Instead, BY color opponent coding feature has a little contribution to discriminate whether face or non-face area. Therefore, in the proposed model, only the real color component R and RG color opponent feature are considered to generate a skin color filter, which also plays a role for reducing computation time as well as getting better skin color filtering performance.

Actually, considering the function of the LGN and the ganglion cells, we implement the on-center and off-surround operation by the Gaussian pyramid images with different scales from 0 to n -th level, whereby each level is made by the sub-sampling of 2^n , thus it is able to construct four feature bases such as the intensity (I), and the edge (E), and color (RG and BY) [11, 12]. This reflects the non-uniform distribution of the retina-topic structure. Then, the center-surround mechanism is implemented in the model as the difference operation between the fine and coarse scales of the Gaussian pyramid images [11]. Consequently, three feature maps are obtained by the following equations.

$$\begin{aligned}
 I(c, s) &= | I(c) \ominus I(s) | \\
 E(c, s) &= | E(c) \ominus E(s) | \\
 RG(c, s) &= | R(c) \ominus G(c) | - | G(s) \ominus R(s) |
 \end{aligned}
 \tag{1}$$

where “ \ominus ” represents interpolation to the finer scale and point-by-point subtraction, c and s are indexes of the finer scale and the coarse scale, respectively. Totally, 18 features are computed because three features individually have 6 different scales [11]. Features are combined into three feature maps as shown in Eq. (2) where \bar{I} , \bar{E} and \bar{C} stand for intensity, edge, and color feature maps, respectively. These are obtained through across-scale addition “ \oplus ” [11].

$$\begin{aligned}
 \bar{I} &= \bigoplus_{c=2}^3 \bigoplus_{s=c+2}^{c+3} N(I(c, s)) \\
 \bar{E} &= \bigoplus_{c=2}^3 \bigoplus_{s=c+2}^{c+3} N(E(c, s)) \\
 \bar{C} &= \bigoplus_{c=2}^3 \bigoplus_{s=c+2}^{c+3} N(RG(c, s))
 \end{aligned}
 \tag{2}$$

Thus, the three features maps such as \bar{I} , \bar{E} and \bar{C} can be obtained by the center-surround difference and normalization (CSD&N) algorithm [11]. A SM is generated by the summation of these three feature maps as shown in Eq. (3).

$$SM = \bar{I} \bar{E} \bar{C}
 \tag{3}$$

The salient areas are obtained by selecting areas with relatively higher saliency in the SM. In order to decide salient area, the proposed model generates binary data for each selected face candidate area using Otsu’s threshold method [12] in the SM. Then, the proposed model makes a group of segmented areas using a labeling method for each binary face candidate area. After obtaining the candidate salient areas for human face, the obtained face candidate areas are used as input of the AdaBoost algorithm.

2.1.1 Facial Feature Detection Using AdaBoost

In case of occluded face, the AdaBoost can not detect the face, because the occluded face does not contain enough facial information. So, we use the facial feature, such as eyes and mouth, to estimate the facial feature area. After detecting the size of eye regions without occlusion, which is obtained by the AdaBoost algorithm [10], we estimate the whole size of the occluded facial area. In this case, since we don’t know whether the detected eye region is for left or right eyes and we need to correctly estimate the whole size of the occluded facial area, we check the energy variation in the face preferable SM. If the energy value in the SM for a left part of detected eye region is greater than that for a right part, then we regard the detected eye as a right eye, and estimate that the occluded facial area is in left side of the right eye. The estimated facial size is obtained by the detected eye region through AdaBoost algorithm and depth information from the binocular saliency.

2.2 Binocular Saliency Map and Depth Information

In order to implement a human-like visual attention function, three processes are integrated to generate a binocular SM. One generates static and dynamic saliency in terms of monocular vision. And we can build a binocular SM by combining two monocular SMs. The stereo vision can get more information from images than monocular vision. Using the binocular SM, we can extract the depth information of each feature. So it can cluster the features by depth information.

First, we have to obtain the degrees of two camera angles to be moved to make a focus on a land mark. Considering the limitation of the field of view (F) in the horizontal axis and motor encoder resolution (U), we can get the total encoder value (E) to represent the limited field of view of the horizontal axis. The total encoder value (E) can be obtained by Eq. (4). As shown in Eq. (5), the total encoder value (E) is used to calculate the encoder value (x_t) of the horizontal axis motor for aligning of each camera to a landmark.

In Eq. (5), R denotes the x -axis pixel resolution of the image and T denotes the relative pixel coordinate of the x -axis of a landmark from the focus position. In other words, T represents the disparity of x - axis. The x - axis encoder value (x_t) that uses to move each camera to the landmark point is translated into the angel (x_d) by Eq. (6). As a result, the angles a and b are obtained by Eq. (6) by substituting T for the x coordinates of the left and right cameras.

$$E=(F \times 360^{\circ}) / U \quad (4)$$

$$x_t = -E + (E \times T) / R \quad (5)$$

$$x_d = 90^{\circ} - (R \times x_t) / U \quad (6)$$

The vertical distance (y) is obtained by the following equations.

$$\tan(a) \times x - \tan(a) \times s = y \quad (7)$$

$$\tan(b) \times x = y \quad (8)$$

$$y = [\tan(a) \times \tan(b)] / [\tan(a) - \tan(b)] \quad (9)$$

Eq. (7) and (8) show the equation of straight lines between the cameras and the landmark, respectively. In Eq. (7), x and y denote the disparities for x -axis and y -axis respectively between a land mark and a current focus position, and s represents the distance between each focal axis of two cameras. Eq. (9) is the equation to calculate the vertical distance (y).

2.3 Occluded Region Detection in Facial Image

We have modeled the occlusion detection mechanism in the IT and V4 areas using AAMLPL by which the characteristics of the facial features are trained and memorized in the connections of the artificial neurons in AAMLPL.



Fig. 2. Example of divided regions

Considering computational efficiency, we extract some eigenvectors with large eigen-values using the principal component analysis (PCA) for extracting important features of a face region. First, we resize detected face into 120 x 120. And we divide face into halves. Lastly, we divide the upper part into halves. The divided each region contains facial features like eye and mouth. Figure 2 shows the divided regions. The PCA features are extracted in each region.

2.3.1 Occluded Region Detection Using AAMLPL

To perceive an occluded facial region, we use the retrieval of face related information from AAMLPL using correlation computation between input and output of the AAMLPL. The AAMLPL has been used successfully in many partially-exposed environments [13]. The face detection is also one of the partially exposed problems with tremendous within-class variability [13]. Let $F(\cdot)$ denotes an auto-associative mapping function, and x_i and y_i indicate an input and output vector, respectively. Then the function $F(x_i)$ is usually trained to minimize the following mean square error given by Eq. (10).

$$E = \sum_{i=1}^n \|x_i - y_i\|^2 = \sum_{i=1}^n \|x_i - F(x_i)\|^2 \quad (10)$$

where n denotes the number of output nodes.

We train the AAMLPL using facial features without occlusion. After detecting a face region, we extract the PCA features at the 3 divided region (2 eyes, 1 mouth), and the facial features in each region are used as input to the each region's AAMLPL as test data. Then, we check whether each region is occluded or not by calculating the mean square error (MSE).

3 Experimental Results

There is no database for experiment of the multiple occluded face detection system. Thus, we make our own database of multiple occlusion using 2 CCD cameras [9]. We get the 200 pictures continuously from 2 cameras. And we use ABR database [14] and POSTECH database [15]. We crop the face from those databases to train the AAMLPL. Totally 93 facial images is trained.

Figs. 3 (a) and (b) show the results of eye detection using AdaBoost algorithm in the localized candidate area. Figs. 3 (c) and (d) show the result of the proposed model, in which the occluded faces are successfully indentified with suitable size of face area. Fig. 4 shows an example for detecting face areas in multiple occluded facial images.

Fig. 3 (e) shows the results of face color preferable attention model in the stereo camera. Fig. 3 (f) shows the results of face and facial feature detection using AdaBoost algorithm in the localized candidate area as shown in Fig. 3 (e). Fig. 3 (g) shows successfully identified with suitable size of face area.

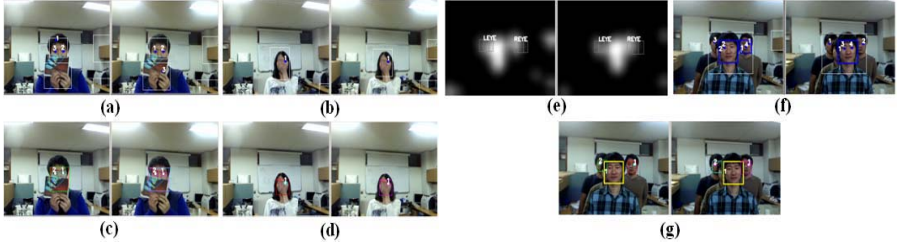


Fig. 3. Occluded face detection results; (a) and (b) Candidate region for face and facial feature detection, (c) and (d) Face detection result for occluded faces, The multiple occluded face detection results; (e) Face preferable SM, (f) Candidate region for face and facial feature detection, (g) Face detection result for multiple occluded faces

Fig. 4 shows the results for detecting the occluded local region using the AAMLPL. We use 264 facial images for test (132 non-occluded face images / 132 occluded face images). From the results, each occluded region has higher MSE than occlusion free region.

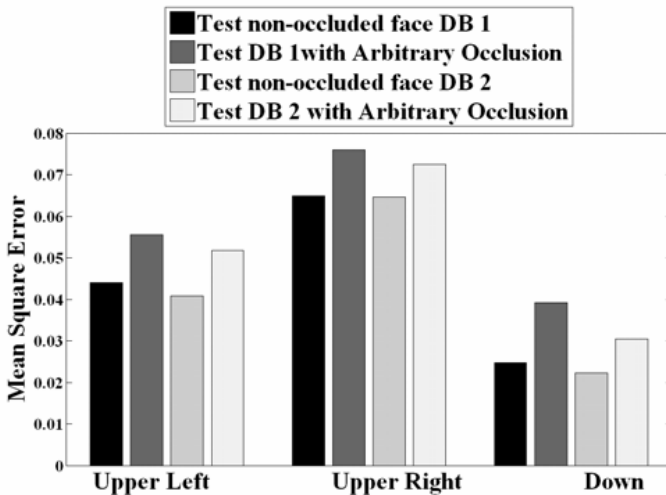


Fig. 4. Test results with average MSW using AAMLPL

4 Conclusion

In this paper, we proposed a novel method which can detect multiple occluded faces and also can detect occluded region in facial image. In natural complex scenes, the proposed

model not only successfully localizes the face areas but also appropriately rejects non-face areas. The proposed model is based on the face color preferable attention, and the AdaBoost algorithm based on Haar-like features decides whether the attended region contains a face characteristic. The proposed model aims to detect multiple occluded faces in crowded area. As a further work, we need to enhance this model and verify the performance of the proposed model through more complex benchmark databases.

Acknowledgments. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2009-0082262).

References

1. Yang, M., Kriegman, D.J., Auja, N.: Detecting faces in images: a survey. *IEEE Trans. Patt. Anal. Mach. Intell.* 20(1), 34–58 (2002)
2. Turk, M.A., Pentland, A.P.: Eigenfaces for Recognition. *J. Cogn. Neurosci.* 3(1), 71–86 (1991)
3. Viola, P., Jones, M.J.: robust real-time face detection. *Int. J. Comput. Vis.* 58(2), 137–154 (2004)
4. Walther, D., Itti, L., Riesenhuber, M., Poggio, T.A., Koch, C.: Attentional selection for object recognition - A gentle way. In: Bülthoff, H.H., Lee, S.-W., Poggio, T.A., Wallraven, C. (eds.) *BMCV 2002. LNCS*, vol. 2525, pp. 472–479. Springer, Heidelberg (2002)
5. Serre, T., Riesenhuber, M., Louie, J., Poggio, T.: On the role of object-specific features for real world object recognition in biological vision. In: Bülthoff, H.H., Lee, S.-W., Poggio, T.A., Wallraven, C. (eds.) *BMCV 2002. LNCS*, vol. 2525, pp. 387–397. Springer, Heidelberg (2002)
6. Navalpakkam, V., Itti, L.: An integrated model of top-down and bottom-up attention for optimal object detection. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2049–2056. IEEE Press, New York (2006)
7. Siagian, C., Itti, L.: Biologically-inspired face detection: Non-Brute-Force-Search approach. In: *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop*, vol. 5, pp. 62–69. IEEE Computer Society, Washington (2004)
8. Ban, S.-W., Lee, M., Yang, H.S.: A face detection using biologically motivated bottom-up saliency map model and top-down perception model. *Neurocomputing* 56, 475–480 (2004)
9. Choi, S.B., Jung, B., Niitsuma, H., Lee, M.: Biologically motivated vergence control system using human-like selective attention model. *Neurocomputing* 69, 537–558 (2006)
10. Kim, B., Ban, S.-W., Lee, M.: Improving adaboost based face detection using face-color preferable selective attention. In: Fyfe, C., Kim, D., Lee, S.-Y., Yin, H. (eds.) *IDEAL 2008. LNCS*, vol. 5326, pp. 88–95. Springer, Heidelberg (2008)
11. Jeong, S., Ban, S.-W., Lee, M.: Stereo saliency map considering affective factors and selective motion analysis in a dynamic environment. *Neural Networks* 21(10), 1420–1430 (2008)
12. Otsu, N.: A threshold selection method from gray-level histogram. *IEEE Trans. System Man Cybernetics.* 9(1), 62–66 (1979)
13. Won, W.J., Jang, Y., Ban, S.: Biologically motivated face selective attention model. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) *ICONIP 2007, Part I. LNCS*, vol. 4984, pp. 953–962. Springer, Heidelberg (2007)
14. Lee, M.: ABR database, <ftp://abr.knu.ac.kr/DB/Occlusion/>
15. Kim, D.: Postech database, <http://imlab.postech.ac.kr>

A Mutual Information Based Face Recognition Method

Iman Makaremi and Majid Ahamdi

Electrical and Computer Engineering Department
University of Windsor
Windsor, ON, Canada
{makarem, ahamdi}@uwindsor.ca

Abstract. A mutual information based method for face recognition has been proposed. By comparing the mutual information of images locally, this method becomes robust to illumination variation. The method's performance has been evaluated using AT&T database with different number of samples in the training set, as well as different resolutions for intensity distribution estimation. The accuracy rate is dependent on the number of samples in the training set and the accuracy of the probability density function (PDF) estimation. An accuracy rate of 94.58% has been obtained when half of the database was used as the training set and the PDFs were estimated with 20-bin histograms. A perfect accuracy rate was achieved when 60% of the database was allocated to the training set.

Keywords: Face Recognition, Mutual Information.

1 Introduction

Face as a non-interactive biometric has been under attention for the last two decades, and face recognition found many applications in security and human-machine interface. There are many challenges in building face recognition systems such as effect of illumination variation, pose, and facial expression.

There are many different approaches to tackle problems raised by illumination variation in face images. Using preprocessing methods which try to enhance the image and remove the effect of illumination [1, 4] is one of them. There are also many other methods that try to solve this problem in feature extraction step. Belhumeur et. al. [3] proposed a method based on Fisher's linear discriminant. Xue et. al. [8] exploited ridge regression locally to make their face recognition method robust to illumination variation. Wright et. al. [7] tried to solve the illumination variation problem as well as facial expression and occlusion with sparse signal representation. Fusing information from different types of sensors is also an alternative. Kong et. al. [4] fused visible and infrared image data to reduce the effect of illumination variation.

In this paper, a face recognition method based on mutual information is introduced. The main advantage of this method is by performing the analysis separately on left and right sides of the images, the effect of side-lighting, which can considerably change the intensity distribution, is reduced significantly. Also, taking advantage of mutual information as a similarity measure between smaller sections of the images makes a local analysis possible and with a simple voting strategy obtains a very high

accuracy face recognition system. In the section 2, the concept of mutual information is briefly reviewed, and the proposed method is explained in detail in section 3. In section 4, experimental results are represented and discussed. Finally, conclusions are presented in section 5.

2 Mutual Information

Mutual information, which is a fundamental concept in information theory [3], is a measure of the dependency between two random variables say X and Y . This dependency is based on the information shared between these two variables, and if they are discrete, it is defined as:

$$I(X; Y) = \sum_x \sum_y p(x, y) \log \frac{p(x, y)}{p_x(x)p_y(y)} \quad (1)$$

where $p(x, y)$ is the joint probability distribution function (PDF) of the variables and $p_x(x)$ and $p_y(y)$ are the PDFs of X and Y respectively.

To estimate the PDFs of images in this paper, we exploit the histogram of intensities of images with different numbers of bins. In this case, the range of 0 to 1 is divided into different numbers of regions with identical width. The effect of number of bins will be discussed in this paper.

3 The Proposed Method

In this method, every image is vertically divided into two sub-images, SI_L and SI_R , with an overlap (Fig. 1-a). The reason is that in the case of side lighting, half of the face is darker than the other side and it directly affects the intensity distribution. If this separation is not done, the intensity distribution will be significantly different from an image with a frontal-lighting. Fig. 2 shows the effect of lighting direction in image histogram (This makes the methods which use the mutual information between the unknown image and images in train set as a primary step to reduce the number of classes for further comparison [5, 6] less reliable). While, dividing the image into two sub-images helps to perform a local analysis on each side which can be made to be more dependent on the general shape of the intensity distribution rather than its location on the intensity axis. This will be explained in the following.

In the next step, each side is divided into smaller overlapping strips (Fig. 1-b). The mean of each strip is moved to 0, and its intensity distribution is estimated. As it was discussed earlier, by moving the mean to 0, we can take advantage of the shape of distribution without being concerned about how the face is lit. Also, because the images are divided into smaller strips, the effect of a sudden change in intensity due to shading cannot considerably affect the shape of distribution. On the other hand, the height of strips should not be very small in order to have a sufficient number of pixels in each of them for accurate intensity distribution estimation. Larger strips also decrease the chance of a localized analysis of variations.

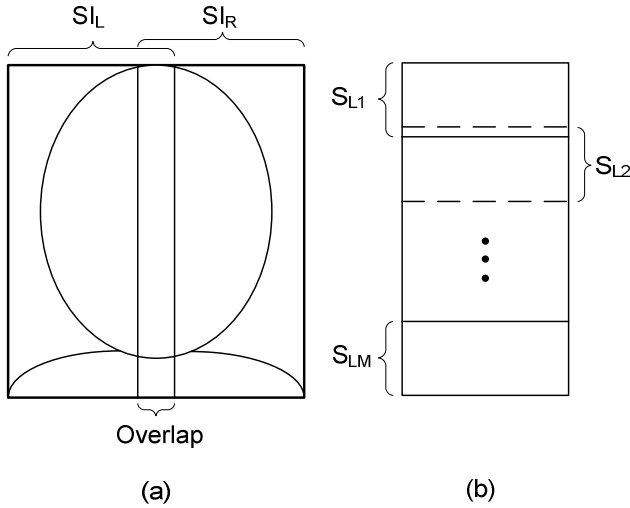


Fig. 1. (a) Each image is divided into two overlapping sub-images. (b) Each sub-image is also divided into overlapping strips.

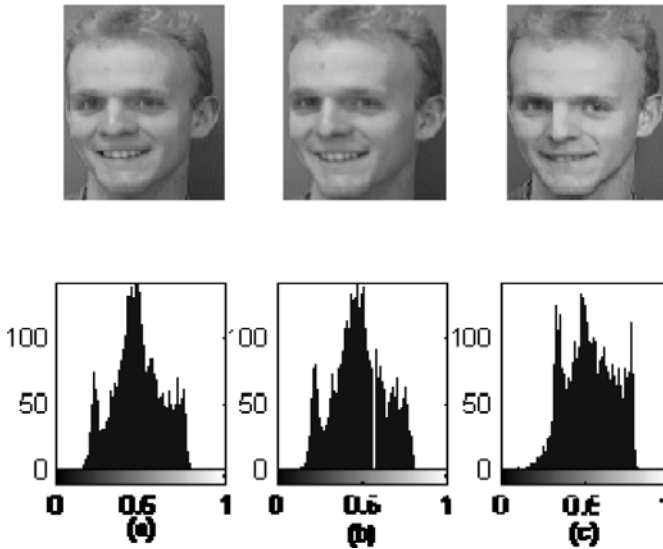


Fig. 2. The effect of lighting direction on image histogram. (a) and (b) have similar histogram, however there is a small pose variation. (c) which has a frontal lighting has a different histogram.

Each image is represented by a set of strips. In this paper, the set of strips of i th image in class c is shown with

$$\mathbb{S}^{c,i} = \{S_{L,j}^{c,i}, S_{R,j}^{c,i} | 1 \leq j \leq M\} \tag{2}$$

where $S_{L,j}^{c,i}$ and $S_{R,j}^{c,i}$ are the j th strips of the left sub-image SI_L and the right sub-image SI_R of the image respectively and M is the number of strips. Mutual information

between \mathbb{S}^x , the representing set of strips of an unknown image in the test set and all $\mathbb{S}^{c,i}$'s of the train set is calculated in order to find the corresponding class. Since the images are divided into strips, comparing strips at similar locations might not include the same parts of face due to movement of camera or of the person (e.g. say eyes might be in the third strip in one image and in the fourth strip in the other one). Therefore, the mutual information between each strip with a certain number, K , of its upper and lower neighbors is also calculated. The mutual information between \mathbb{S}^x and $\mathbb{S}^{c,i}$ with k shifts is shown as follows:

$$I_k^{c,i} = MI(\mathbb{S}_k^x, \mathbb{S}^{c,i}) \quad (3)$$

where MI is the mutual information function, and the subscript k shows the number of shifts. For each shift, there are $2M$ (the number of strips on both sides) mutual information values. The average of these values is used as the representative mutual information between two images.

Between $2K+1$ shifts, the highest mutual information represents the similarity of the two images:

$$I^{c,i} = \max_{-K \leq k \leq K} (I_k^{c,i}) \quad (4)$$

which is used to determine the class of the image. If there is more than one image per class in the train set (say N), the average of the $I^{c,i}$'s are calculated as the similarity between the image and that certain class:

$$I^c = \frac{\sum_{i=1}^N I^{c,i}}{N}. \quad (5)$$

Finally, the image is recognized to belong to a class which it had the highest similarity with:

$$class = \arg \max_c (I^c). \quad (6)$$

4 Experimental Results

In this paper, we have used AT&T face database. This database contains 400 different images of 40 individuals, 10 for each. The size of images is 112×92 . The images were divided into two sub-images with a width of 50 pixels; so they had overlap on 8 pixels. Afterward, the height of strips was set to 8 pixels, and the overlap between them was 5 pixels. Therefore, there were 400 pixels in each strip to estimate their PDF. The PDFs were defined as the histogram of the intensities of the strip with different numbers of bins.

To find the best match, the number of neighbor strips on each side, K , was set to 2. Thus, displacements of up to 6 pixels on each side were detectable. In this study, the effect of two different parameters on the classification rate has been studied; 1) the number of bins for PDF estimation, 2) number of samples in training set.

The algorithm has been implemented in MATLAB and executed sixty times. The averages of the results are shown in Table I and Fig. 3. Based on these results, the number of bins has a significant effect on the classification rate. In this study, the best results were obtained with 20 bins. The reason is that, small number of bins does not give an accurate estimation of the PDFs, and a larger number of bins makes the

distribution to be too detailed which has a negative effect on the accuracy. Fig. 4 illustrates a strip and its histograms with six different numbers of bins. The strip shows a part of the face right below the left eye of the person. As it is shown, the histogram with 5 bins shows a very general shape of the distribution while at the other end the histogram with 100 bins represents the strip with lots of variations which makes it very sensitive to small variations in illumination. Based on Table 1, small numbers of bins have very poor results, and the results with larger number of bins are also unacceptable.

The best results were obtained when the number of bins was 20. The accuracy rate with one sample in training set is 48.72%, and it increases to 94.58% with 5 samples and 100% for more samples.

Table 1. Accuracy rates based on Number of Bins (NoB) and Number of Samples (NoS) in train set

NoB \ NoS	5	10	20	30	50	100
1	20.18	46.00	48.72	50.48	26.85	2.11
2	22.25	56.89	64.27	60.66	26.79	2.21
3	28.23	72.13	75.45	65.30	27.71	2.39
4	30.56	84.10	85.52	69.38	26.95	2.50
5	34.90	89.95	94.58	70.88	26.91	2.45
6	38.44	91.69	100.00	72.51	27.11	2.46
7	42.33	93.17	100.00	73.57	26.43	2.50
8	43.69	98.94	100.00	74.58	27.20	2.50
9	46.13	100.00	100.00	74.85	26.10	2.50

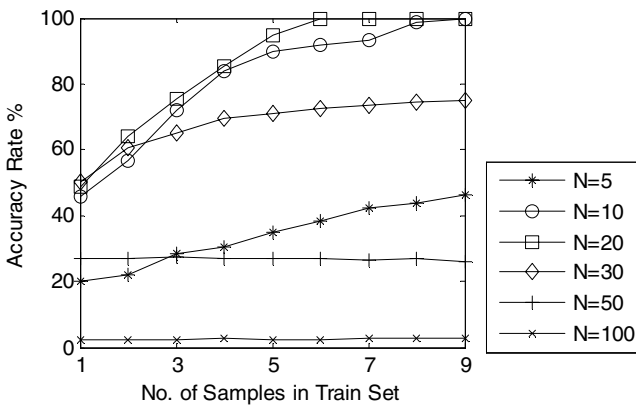


Fig. 3. Accuracy Rate with different number of Samples in train set and different number of bins

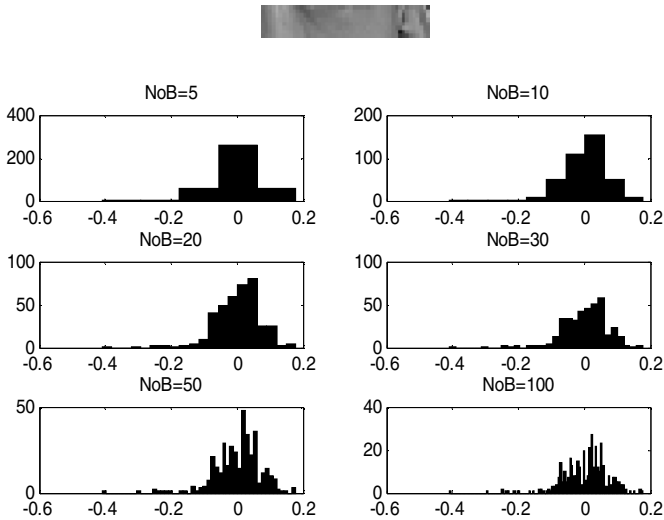


Fig. 4. A strip of face and histograms of it with different numbers of bins

5 Conclusion

A face recognition method which uses mutual information as a measure of similarity between images was proposed. In this method, images were divided into smaller overlapping strips, and mutual information between these strips and their corresponding strips in other images were calculated. Also, to reduce the effect of illumination variation in different images, the mean of each strip was moved to zero. Considering the possibility of displacement of camera and/or face in images, the mutual information between strips and a certain number of their corresponding neighbors in other images were calculated in order to find the best match. In this paper, the histogram of each strip was used to estimate the intensity distribution. To have a better understanding of the estimation, different number of bins were used. Also, the effect of different number of samples in train set was studied. The accuracy rate on AT&T face database while half of the database was used as the train set and with 20-bin histograms was %94.58. The accuracy rate rises to 100% when 60% or more of the database is allocated to the training set with the same number of bins.

References

1. Arandjelovic, O., Cipolla, R.: A methodology for rapid illumination-invariant face recognition using image processing filters. In: *Computer Vision and Image Understanding*, February 2009, vol. 113, pp. 159–171 (2009)
2. Belhumeur, P.N., et al.: Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 711–720 (1997)
3. Cover, T.M., Thomas, J.A.: *Elements of Information Theory*. Wiley, Chichester (1991)

4. Singh, R., et al.: Improving verification accuracy by synthesis of locally enhanced biometric images and deformable model. *Signal Processing* 87, 2746–2764 (2007)
5. Su, H.-T., Feng, D.-., Wang, X.-Y., Zhao, R.-C.: Face recognition using hybrid feature. In: 2003 International Conference on Machine Learning and Cybernetics, vol. 5, pp. 3045–3049 (2003)
6. Hongtao, S., Feng, D.D., Rong-chun, Z., Xiu-ying, W.: Face recognition method using mutual information and hybrid feature. In: Proceedings of Fifth International Conference on Computational Intelligence and Multimedia Applications, ICCIMA 2003, pp. 436–440 (2003)
7. Wright, J., et al.: Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 210–227 (2009)
8. Xue, H., et al.: Local ridge regression for face recognition. *Neurocomputing* 72, 1342–1346 (2009)

Basis Selection for 2DLDA-Based Face Recognition Using Fisher Score

Peratham Wiriyathammabhum and Boonserm Kijirikul

Department of Computer Engineering, Faculty of Engineering
Chulalongkorn University
Bangkok 10330 Thailand

g52pwr@cp.eng.chula.ac.th, Boonserm.k@chula.ac.th

Abstract. Two-Dimension Linear Discriminant Analysis (2DLDA) becomes a popular technique for face recognition due to its effectiveness in both accuracy and computational cost. Furthermore, there has been shown that 2DLDA reduces only the row direction of the data. This gives a rise to a new technique, $(2D)^2$ LDA. $(2D)^2$ LDA performs 2DLDA on the row direction and conducts Alternate 2DLDA on the column direction of the data. Although the eigenvalues associated with eigenvectors simply show the discriminative power of the subspace spanned by the corresponding eigenvectors, there are some evidences indicate the eigenvector with high eigenvalue may correspond to noise signal such as pose, illumination or expression and the eigenvector with high discriminative power may have a low eigenvalue due to its closeness to the null space of the training data. By these reasons, we may improve the performance of 2DLDA-based techniques by properly reordering the importance of their eigenvectors. In this paper, we propose a technique to solve this problem; we use the Subspace Scoring with the Fisher Criterion to rerank the discriminative power of the subspace spanned by certain eigenvectors. The experimental results show that our method makes an improvement to 2DLDA and $(2D)^2$ LDA in accuracy. We also combine our proposed method with the wrapper method to determine the target dimension for further use.

Keywords: Fisher Score, face recognition, Linear Discriminant Analysis, LDA, 2DLDA, $(2D)^2$ LDA, wrapper.

1 Introduction

For the last decades, many research works have proved the effectiveness of feature extraction techniques in face recognition. They assumed that the face data can be represented in the underlying intrinsic dimension. Two dimension Linear Discriminant Analysis (2DLDA) [3] is one of the efficient techniques which is developed from LDA [1]. It directly uses the image matrix representation of the data which better preserves the local information between pixels of the image data. The scatter matrices are much smaller than the original version. This makes it require less time and memory in computation process. 2DLDA was also reported to be more accurate due to more information preserved in the computation.

Later, there has been emphasized that 2DLDA reduces only the dimension of the row direction. This leads to the new techniques, (2D)²LDA [4]. (2D)²LDA reduces both row and column directions of the data simultaneously by using the Alternate version of 2DLDA to obtain another projection matrix.

This paper focuses on two LDA-based techniques, 2DLDA and (2D)²LDA and emphasizes on the phenomenon happened in random subspace methods reported in the work of Nyugen et al. [5]. The phenomenon is about the random selection of basis vectors may form a more discriminative subspace and yield a better result in classification accuracy. Nyugen et al. conducted the research on 2DPCA but we investigate on 2DLDA instead. For LDA, there are some works [2, 6] on the issue of a discriminative basis that has lower weight in choosing for a candidate basis remained in dimensional reduction. The first work shows that the basis with largest corresponding eigenvalue is affected by illumination condition. Another work is on the singularity problem of LDA that was completely overcome in 2DLDA, but there may be some trace of this problem that may also reside in 2DLDA.

The organization of this paper is as follows. The overview of 2DLDA and (2D)²LDA are in Section 2. The overview of Fisher Score is in Section 3. The proposed method is in Section 4. The experimental results are in Section 5. Finally, the paper conclusion is in Section 6.

2 Overview of 2DLDA and (2D)²LDA

2.1 Notations

We denote data with labels by $\{A_i, y_i\}$, $y_i \in \{1, 2, \dots, c\}$ where A_i is the matrix representation by the image representation of data. Let μ , n_i , μ_i , σ_i^2 , P and Z denote the global mean value, the number of data in class i , the mean value in class i , the variance in class i , the projection matrix that has basis vector φ in column manner, $P = [\varphi_1 | \varphi_2 | \dots | \varphi_n]$ and the feature matrix after dimensional reduction, respectively.

Frobenius norm is the matrix norm; for example, Frobenius norm of matrix $A-B$ is denoted as $\|A - B\|_F$ which can be calculated as:

$$\|A - B\|_F = \sqrt{\sum_{i,j} \|a_{ij} - b_{ij}\|_2^2}, \tag{1}$$

where $\|a_{ij} - b_{ij}\|_2$ denotes the Euclidean norm or L2 norm.

2.2 2DLDA

2DLDA [3] is a popular and effective technique for face recognition. 2DLDA uses the image representation which is a matrix or the mode-2 tensor instead of the traditional representation of vector in the classic LDA [1]. This makes the construction of the scatter matrix more accurate due to lower computation which also reduces the time.

2DLDA tries to maximize the Fisher's criterion and finds an optimal projection U using dimension-reduction equation $Z = U^T A$. The criterion is as follows:

$$U_{opt} = \max_U \frac{|U^T S_b U|}{|U^T S_w U|}. \tag{2}$$

We define M as a global mean calculated from the Frobenius norm of data matrices A , and M_i as a within class mean of data matrices A of class i .

$$S_b = \sum_{i=1}^c n_i (M_i - M)(M_i - M)^T \tag{3}$$

$$S_w = \sum_{i=1}^c \sum_{x_k \in X_i} (A_k - M_i)(A_k - M_i)^T. \tag{4}$$

The optimal projection matrix U can be calculated by solving the eigenvalue problem of matrix $(S_w)^{-1}S_b$. Note that it has been shown that matrix S_w is always nonsingular because the number of data matrix A will always exceed the rank of A [3]. U will be formed as $[\varphi_1|\varphi_2|\dots|\varphi_p]$, where φ are the eigenvectors sorted by the corresponding eigenvalues λ in descending order and p is from the notation $p \times q$ row by column representation of image matrix A . We should choose $d_1 < p$ for the target reduced dimension. This makes projection matrix U be $U_{d1} = [\varphi_1|\varphi_2|\dots|\varphi_{d1}]$.

2.3 Alternate 2DLDA

Alternate 2DLDA [4] is an alternate version of 2DLDA. It operates on the column direction of image matrix A . Alternate 2DLDA finds an optimal projection V using dimension-reduction equation $Z = AV$. The criterion is the same as 2DLDA. The difference between Alternate 2DLDA and 2DLDA is the way of constructing between class scatter matrix S_b and within class scatter matrix S_w :

$$S_b = \sum_{i=1}^c n_i (M_i - M)^T (M_i - M) \tag{5}$$

$$S_w = \sum_{i=1}^c \sum_{x_k \in X_i} (A_k - M_i)^T (A_k - M_i). \tag{6}$$

The optimal projection matrix V can be calculated by solving the eigenvalue problem of matrix $(S_w)^{-1}S_b$. V will be formed as $[\varphi_1|\varphi_2|\dots|\varphi_q]$, where φ are the eigenvectors sorted by the corresponding eigenvalues λ in descending order and q is from the notation $p \times q$ row by column representation of image matrix A . We should choose $d_2 < q$ for the target reduced dimension which makes projection matrix V be $V_{d2} = [\varphi_1|\varphi_2|\dots|\varphi_{d2}]$.

2.4 (2D)²LDA

We show in Section 2.3 that 2DLDA works in the row direction to reduce the image matrix A of $p \times q$ elements to feature matrix Z of $d_1 \times q$ elements. Similarly, we also show in Section 2.4 that Alternate 2DLDA works in the column direction which alternatively reduces dimensions of $p \times q$ image matrix A to $p \times d_2$ feature matrix Z . Suppose we calculate both 2DLDA and Alternate 2DLDA with the training set of image matrix A and we obtain the two projection matrices U and V . If we perform the dimension reduction simultaneously, the equation $Z = U^T A$ and $Z = AV$ can be

combined to a new equation to form a new feature matrix $Z = U^TAV$ with dimensions $d_1 \times d_2$. In conclusion, (2D)²LDA [4] performs 2DLDA and Alternate 2DLDA to yield projection matrices U and V for the equation $Z = U^TAV$ and (2D)²LDA will have two feature dimension parameters d_1 and d_2 which makes the dimension-reduction equation to be $Z = U_{d_1}^TAV_{d_2}$.

2.5 Classification Method of 2DLDA-Based Techniques

We will classify the data using the similarity measured from distance between their feature matrices. Large distance means low similarity. For 2DLDA and Alternate 2DLDA, we denote two feature matrices $Z_1 = [z_{11}; z_{12}; \dots; z_{1d}]$ and $Z_2 = [z_{21}; z_{22}; \dots; z_{2d}]$ the similarity of Z_1 and Z_2 can be calculated as follows:

$$d(Z_1, Z_2) = \sum_{k=1}^d \|z_{1k} - z_{2k}\|_2. \quad (7)$$

For (2D)²LDA, we calculate the similarity of two feature matrices from their difference on Frobenius norm,

$$d(Z_1, Z_2) = \|Z_1 - Z_2\|_F. \quad (8)$$

3 Fisher Score

Fisher Score [9] is a supervised feature selection technique in the category of filter methods. It is sometimes denoted as Fisher Kernel and widely used as a kernel or used with the Hidden Markov Model (HMM). It selects a good feature by the score that is measured by its discriminative power defined by Fisher's Criterion. Given data with labels $\{x_i, y_i\}$, $y_i \in \{1, 2, \dots, c\}$. F denotes the Fisher Score value, μ denotes the global mean value, n_i denotes the number of data in class i and μ_i denotes the mean value in class i . σ_i^2 denotes the variance in class i . Fisher Score criterion is as follows:

$$F = \frac{\sum_{i=1}^c n_i (\mu_i - \mu)^2}{\sum_{i=1}^c n_i \sigma_i^2}. \quad (9)$$

Fisher Score directly measures the value F for each feature. A feature will have a high score if it has high between class scatter and low within class scatter. This can be seen that Fisher Score indicates the discriminative value on one feature which is in a form of vector only.

4 Proposed Method

It is obvious that the eigenvectors obtained from 2DLDA, Alternate 2DLDA and (2D)²LDA may not have their corresponding discriminative power in the decreasing order of their corresponding eigenvalues. The first reason is when training data to obtain the model, 2DLDA based methods may classify the training data using unimportant features in the data such as pose, illumination or expression [2]. This will

make us obtain the wrong model for testing. The second reason is that sometimes the most discriminative projection may be an eigenvector with a low eigenvalue due to its closeness to the null space [6].

We also assume that all obtained eigenvectors are distinct in discriminative power and we want to reorder the eigenvectors in order to form a correct subset and select them for the dimensional reduction step. To achieve this target, we propose the use of Fisher Score which can evaluate the discriminative power in the vector space obtained in the dimensional reduction step of eigenvectors with low computational cost of $O(n)$ where n is the number of training data.

For 2DLDA and Alternate 2DLDA, we denote the projection matrix U and V as $P=[\varphi_1|\varphi_2|\dots|\varphi_r]$, where φ are the eigenvectors sorted by the corresponding eigenvalues λ in descending order. We will obtain the corresponding vector space S of eigenvectors φ and the data X from the equation $S = \varphi^T X$ as shown in Fig.1. S has the dimension $r \times 1$. Then, we evaluate the Fisher Scores of all vector spaces S_i , namely, $S_i = \varphi_i^T X$. Then, we get Fisher Score F_i corresponding to vector space S_i spanned by eigenvector φ_i . Finally, we rearrange the set of eigenvectors $P_{fs}=[\varphi_1|\varphi_2|\dots|\varphi_r]$ in the decreasing order of Fisher Score F_i . The computational complexity in Fisher Score for 2DLDA and Alternate 2DLDA is $O(pn)$ and $O(qn)$ from the notation $p \times q$ row by column representation of image matrix A .

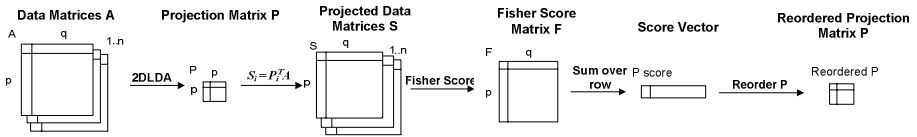


Fig. 1. Thumbnail of the 2DLDA with Fisher Score algorithm

For $(2D)^2LDA$, we perform traditional 2DLDA and Alternate 2DLDA, and then use two vectors from the obtained projection matrix U and V to form the subspace matrix S with dimension $p \times q$ as shown in Fig.2. Then, we calculate the Fisher Score matrix F from each cell of matrix S and the row summation of F is the Fisher Score corresponding to each vector in projection matrix U . Similarly, the column summation of F is the Fisher Score corresponding to each vector in projection matrix V .

After the evaluation of Fisher Score, we may use the wrapper method to determine the optimal target dimension. Although Fisher Score can rearrange the discriminative power of eigenvectors, it is still impractical to pick up an optimal target dimension d in 2DLDA and Alternate 2DLDA or d_1 and d_2 in $(2D)^2LDA$. We may define a cutoff c based on Fisher Score itself but it is also difficult to choose the optimal c .

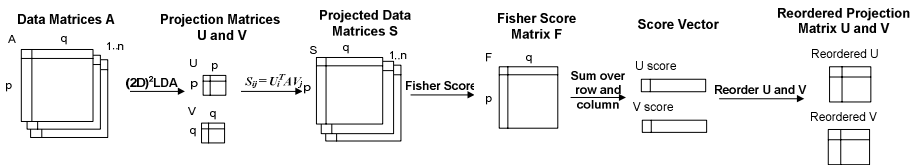


Fig. 2. Thumbnail of the $(2D)^2LDA$ with Fisher Score algorithm

Wrapper methods [10] are feature selection methods which wrap the classifier within the evaluation step as an inductive algorithm. The classifier is used with the cross-validation strategy, which holds out a subset of the whole dataset to be used as a validation set, to evaluate the classification error. The wrapper methods determine the optimal subset of input features with the searching techniques. In this paper, we need an optimal solution regardless of training time so we select the exhaustive search with pruning to reduce some training time.

The exhaustive search normally has a search space of $O(2^d)$ where d is the number of feature vectors when a wrapper method is used to evaluate all subsets from the combination of eigenvectors. However, if we use the filter method, such as Fisher Score, for preprocessing, the exhaustive search can be used with the subsets obtained from the ordered combination of eigenvectors instead, ie. $\{[\varphi_1], [\varphi_1|\varphi_2], \dots, [\varphi_1|\varphi_2|\dots|\varphi_d]\}$. This will reduce the subspace from $O(2^d)$ to $O(d)$. In case of $(2D)^2$ LDA, it will be reduced from $O(2^{d_1+d_2})$ to $O(d_1 \times d_2)$. We also use pruning to reduce the computation time.

5 Experimental Results

We evaluated our proposed methods and their originals on the Yale face database¹ and Extended Yale face database B² [7, 8]. For Yale face database, we evaluated with the subset of 15 individuals with 11 images per person on different face expressions and some configurations such as glasses. For Extended Yale face database B, we evaluated with the subset of 38 individuals with 64 near frontal images on different illuminations. In both data sets, each image was cropped and rescaled to 32 by 32. In our experiments, the data set was randomly partitioned into 10 roughly equal-sized subsets. Each subset was used as a test once, and the remaining subsets were used as the training set. In the experiments, 1-Nearest Neighbor classifier (1-NN) was employed.

5.1 Original 2DLDA and $(2D)^2$ LDA vs. 2DLDA and $(2D)^2$ LDA with Fisher Score

In this section, we tested the improved performance when applying the Fisher Score for basis selection to 2DLDA and $(2D)^2$ LDA. The dimensions to be reduced are vary from 1 to 32 according to the number of rows in the image matrix. This test was carried on Intel Core 2 Duo CPU 2.2 GHz., Windows Vista 32 bits OS and 2 GB RAM. Table 1 shows that our proposed method improves the performance of 2DLDA by 0.67% and 1.55% and $(2D)^2$ LDA by 2% and 5.42% on both database respectively. This indicates that the performances of 2DLDA-based techniques are improved by reordering the eigenvectors using the Fisher Score. It is known that $(2D)^2$ LDA does not always outperform 2DLDA or may slightly outperform 2DLDA only but the results show that it can significantly outperform 2DLDA. From the assumption of eigenvectors misarrangement, $(2D)^2$ LDA uses two projection matrices which can be significantly suffered from the error caused by each projection matrix. Our proposed method gives the thorough measurement of how to rearrange the eigenvectors and gives us an unleashed performance of $(2D)^2$ LDA.

¹ <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>

² <http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html>

To compare with the state-of-the-art Fisherface, our method was divided into two stages, similar to Fisherface which is a two-stage PCA+LDA. In the first stage, we reduced the dimension to the cutoff of 97% of cumulative Fisher Score. In the second stage, we performed LDA to reduce the desired dimension to be less than the number of classes. The results show that in the small data set, Yale face database, our method outperforms Fisherface and in the large data set, Extended Yale Face database B, our method + LDA is comparable with the Fisherface. This means our method performs well on both scenarios of small and large data sets.

Table 1. Comparisons of the best classification accuracy of the original 2DLDA and $(2D)^2$ LDA with 2DLDA and $(2D)^2$ LDA with the Fisher Score using 1-NN

Technique	Yale face database	Extended Yale face database B
2DLDA	82.33%	90.40%
2DLDA with Fisher Score*	83.00%	91.95%
$(2D)^2$ LDA	84.33%	90.07%
$(2D)^2$ LDA with Fisher Score*	86.33%	95.49%
Fisherface (PCA+LDA)	78.67%	97.24%
$(2D)^2$ LDA with Fisher Score + LDA*	79%	97.67%

* indicates our proposed methods.

5.2 $(2D)^2$ LDA and $(2D)^2$ LDA with Fisher Score and Wrapper

In this section, we picked up the high accuracy $(2D)^2$ LDA to choose the optimal target dimension for real world application where the optimal one is unknown. We combined the use of wrapper with Fisher Score to determine the optimal subset of bases. In the wrapper phase, we used 10-fold cross validation as an evaluation function and apply some pruning to reduce computation time. The environment was the same as the previous sub-section. We evaluated on Yale face database which has smaller size and the classification accuracies are in Table 2. The results indicate that our proposed method helps improve the classification accuracy of $(2D)^2$ LDA and this combination extends the use of $(2D)^2$ LDA in a real world problem.

Table 2. Comparisons of Classification Accuracy of the original 2DLDA and $(2D)^2$ LDA with 2DLDA and $(2D)^2$ LDA with the Fisher Score using 1-NN

Technique	Yale face database
$(2D)^2$ LDA with wrapper	75.00%
$(2D)^2$ LDA with Fisher Score and wrapper*	77.67%

6 Conclusions

In this paper, we visited the problem of the eigenvector misarrangement in 2DLDA and $(2D)^2$ LDA which causes the degradation in discriminative power of a dimensional

reduced subspace. Our proposed method picks up a misplaced eigenvector that also has a high discriminative power but has a low corresponding eigenvalue to yield a more proper order when selecting the subset eigenvectors as a projection matrix for the dimensional reduction step. The experimental results show that our proposed method helps unleash the performance of 2DLDA and significantly in $(2D)^2$ LDA where the original technique seems to be more suffered from the misarrangement problem. Our proposed method requires an $O(n)$ computational cost which makes it practical in applications. We also proposed the combination of filter and wrapper method to tackle this problem in real world applications. However, the guarantee of the global optimum in the unleash performance is still unclear and it is unknown that if there exists any better scoring way than Fisher Score and the computational cost to find the target dimension is still costly. These will be investigated in our future works.

References

1. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, London (1991)
2. Belhumeur, P.N., Hespanha, J., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Trans. Pattern Anal. Mach. Intell.* 19(7), 711–720 (1997)
3. Ming, L., Yuan, B.: 2D-LDA: a statistical linear discriminant analysis for image matrix. *Pattern Recognition* 26(5), 527–532 (2005)
4. Noushath, S., Hemantha Kumar, G., Shivakumara, P.: (2D)²LDA: An efficient approach for face recognition. *Pattern Recognition* 39, 1396–1400 (2006)
5. Nguyen, N., Liu, W., Venkatesh, S.: Random Subspace Two-Dimensional PCA for Face Recognition. In: Ip, H.H.-S., Au, O.C., Leung, H., Sun, M.-T., Ma, W.-Y., Hu, S.-M. (eds.) *PCM 2007*. LNCS, vol. 4810, pp. 655–664. Springer, Heidelberg (2007)
6. Chen, L., Liao, H., Ko, M., Lin, J., Yu, G.: A New LDA based Face Recognition System Which can Solve the Small Sample Size Problem. *Pattern Recognition* 33(10), 1713–1726 (2000)
7. Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. *IEEE Trans. Pattern Anal. Mach. Intelligence* 23(6), 643–660 (2001)
8. Lee, K.C., Ho, J., Kriegman, D.: Acquiring Linear Subspaces for Face Recognition under Variable Lighting. *IEEE Trans. Pattern Anal. Mach. Intelligence* 27(5), 684–698 (2005)
9. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer, Heidelberg (2006)
10. John, G.H., Kohavi, R., Pflieger, K.: Irrelevant Features and the Subset Selection Problem. In: *International Conference on Machine Learning*, pp. 121–129 (1994)

A Robust Keypoints Matching Strategy for SIFT: An Application to Face Recognition

Minkook Cho and Hyeyoung Park

The School of Electrical Engineering and Computer Science,
Kyungpook National University of South Korea
ucaresoft@paran.com, hyepark@knu.ac.kr

Abstract. Recently, the Scale Invariant Feature Transform (SIFT) proposed by Lowe has emerged as a cut edge methodology in general object recognition as well as for other machine vision applications. However, SIFT method has not shown successful results in face recognition problem because of its original matching strategy which does not consider the location of local keypoints. This paper proposes a novel keypoints matching strategy for face recognition. The proposed matching strategy can avoid mis-matching of local keypoints by using regular grid of face image and can give robustness to various transformations by using keypoint voting strategy. By performing computational experiment on the AR face data set, we confirmed the proposed matching strategy gives better performance than the conventional methods. Especially, the proposed method can give robust and best performance for facial images with occlusions.

Keywords: Scale Invariant Feature Transform (SIFT), face recognition, matching strategy.

1 Introduction

Face recognition has become a very active research topic in last decade [1,2]. In pattern recognition and computer vision domain, several and novel approaches have been introduced for face recognition in recent literatures. Still, face recognition is a difficult challenge since human face is not rigid object and can be transformed easily under different environments. Therefore, in order to increase the accuracy of face recognition systems, it is very important to find an efficient representation of human face which can give clear distinction between subjects.

One of most well-known representation methods are PCA (Principal Component Analysis) and LDA (Linear Discriminant Analysis), which utilize a linear transformation matrix to obtain meaningful features satisfying specific conditions. PCA [3,4,5] computes a reduced set of orthogonal basis vectors or eigenfaces of the training face images. A new face image can be approximated by a weighted sum of these eigenfaces. It provides an optimal linear transformation from the original image space to an orthogonal eigenspace with reduced dimensionality in the sense of least mean squared reconstruction error. LDA [6,7,8]

seeks to find a linear transformation by maximizing the between-class variance and minimizing the within-class variance. However, these methods hardly give a robust performance to face recognition because of various conditions such as illumination, occlusion, or expression changes.

Scale Invariant Feature Transform (SIFT) [9,10] proposed by Lowe becomes one of the popular feature extraction method for pattern recognition because of the excellent performances shown in the object recognition problem. The SIFT method first detects local keypoints that are notable and stable for images in different resolutions, and uses scale and rotation invariant descriptors to represent the keypoints. However, SIFT method has rarely been applied to face recognition because of its matching strategy. The original matching strategy proposed by Lowe is to find the best candidate match for each keypoint by identifying its nearest neighbor in database of keypoints from each training image. In this matching strategy, the location of features is not considered, which may cause severe problems.

To apply SIFT for face recognition, the SIFT-Grid method [11] is introduced by Bicego and et al. The SIFT-Grid method first conducts the conventional SIFT on whole face image and divides it into regular grids. Finally, it conducts keypoints matching in each subregion. Representation of an image using a combination of subregions can give some locational information of local keypoints.

In this paper, by extending the SIFT-Grid method, we propose a novel keypoints matching strategy for face recognition. The proposed matching strategy can avoid mis-matching of local keypoints by using regular grid of face image which gives locational information of local keypoints. Also the proposed matching strategy can give robustness to various transformations by using keypoint voting strategy which utilize the local independent property of keypoints.

The conventional SIFT method is demonstrated in Section 2, and the proposed keypoints matching strategy is introduced in Section 3. The experimental results are shown in Section 4. Finally, conclusions are made in Section 5.

2 Scale Invariant Feature Transform

Scale Invariant Feature Transform has been proposed for extracting distinctive invariant features from images, which can be used to perform reliable matching between different views of an object or scene. It consists of two main stages of computation to generate the set of image features [10].

- 1. Keypoint detector: The invariant feature to scale and orientation is detected by using a difference-of-Gaussian function. The difference-of-Gaussian function is as follows:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (1)$$

where $I(x, y)$ and $L(x, y, \sigma)$ represent a scale-space function and a convolution function respectively and k is multiple factor. Then local maxima and

minima of $D(x, y, \sigma)$ are computed based on its eight neighbors in current image and nine neighbors in the scale above and below. The gradient magnitude $m(x, y)$ and orientation $\Theta(x, y)$ are also computed as follows:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2)$$

$$\Theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1))/(L(x+1, y) - L(x-1, y))). \quad (3)$$

From the obtained local maxima and minima, keypoints are selected based on measures of their stability, gradient magnitude $m(x, y)$ and orientation $\Theta(x, y)$.

- 2. Keypoint descriptor: The obtained keypoint is composed by four part: the locus (location in which the feature has been found), the scale, the orientation and the descriptor. The feature descriptor, which is represented by a 128-dimensional vector, is obtained by computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location weighted by a Gaussian window. For simplicity, given a keypoint κ_i , let us denote $F(\kappa_i), L(\kappa_i), S(\kappa_i), O(\kappa_i)$ as its feature descriptor, location, scale, and orientation, respectively.

The SIFT method detects the local keypoints that are notable and stable for images in different resolutions and uses scale and rotation invariant descriptors to represent the keypoints. However, the naive matching strategy of SIFT which finds the minimum pair distance (we will describe in next section in detail) for keypoints is not suitable to face recognition. Therefore, in next section, we will introduce new matching strategies to overcome this drawback.

3 Keypoints Matching Strategies

3.1 Conventional Matching Strategies

Before the keypoints matching, we should conduct SIFT on training data set and test data set. The original SIFT should match a test image with each training image. For a training image and a test image, the obtained keypoints are represented by:

$$K(I_{test}) = \{\kappa_1^{I_{test}}, \kappa_2^{I_{test}}, \dots, \kappa_M^{I_{test}}\}, \quad (4)$$

$$K(I_{train}) = \{\kappa_1^{I_{train}}, \kappa_2^{I_{train}}, \dots, \kappa_N^{I_{train}}\}, \quad (5)$$

where M and N denote the number of obtained keypoints for each image respectively. The minimum pair distance is computed as follows:

$$D(I_{test}, I_{train}) = \min_{i,j} (d(F(\kappa_i^{I_{test}}), F(\kappa_j^{I_{train}}))) \quad (6)$$

where $d(F(\kappa_i), F(\kappa_j))$ is a distance between descriptors. In the original matching strategy using this minimum pair distance, the locations of features are not considered, which may cause a severe problem of locational mis-matching. The

matched keypoint pair which gives minimum distance, may come from strictly different parts of facial images. For example, the keypoint of left eye may be mis-matched with one of right eye or mouth. This kind of mis-match can cause low classification performance.

To overcome this drawback, SIFT-Grid proposed by Bicego and et al. [11], divides the images into a number of sub-images using a regular grid with overlapping. The distance between two images can be measured by computing minimum distance between all pairs of corresponding sub-images and averaging them as follows:

$$D^{RG}(I_{test}, I_{train}) = \frac{1}{T} \sum_{t=1}^T (D(I_{test}^t, I_{train}^t)) \quad (7)$$

where I^t denotes t th partial overlapped sub-image and T denotes the number of sub-images.

Though the size of sub-image may depend on data, this paper uses 1/4 and 1/3 of width and height, respectively. Using combination of sub-images, we can expect to avoid the locational mis-matching of keypoints which often occurs in the original matching strategy.

3.2 Keypoint Voting

The standard SIFT method and SIFT-Grid method try to measure the distance between test image and each training image so as to assign the test image to a class from which the training image with minimum distance is given. In the proposed method, however, we utilize the independent properties of local keypoints given by SIFT and try to assign each keypoint in the test image to a specific class independently. The result of assignment for each keypoint will play a vote in the decision of the class membership of the whole test image.

In the proposed keypoint voting method, we first construct a keypoint-pool from all training image as follows:

$$\text{Keypoint-pool} = \{\kappa_j^i \mid i = 1, \dots, N, j = 1, \dots, N_i\} \quad (8)$$

where N represents the number of training images, N_i represents number of obtained keypoints for the i th image and κ_j^i denotes j th keypoint of i th training image. The minimum pair distance between k th keypoint of a test image and the keypoint-pool is computed as follows:

$$D(\kappa_k^{test}, \text{Keypoint-pool}) = \min_{i,j} (d(F(\kappa_k^{test}), F(\kappa_j^i))) \quad (9)$$

where κ_k^{test} represents k th keypoint of the test image. Based on the distance, we can obtain the class-label that the keypoint indicates, such as

$$C(\kappa_k^{test}) = C(\arg \min_{\kappa_j^i} \{d(F(\kappa_k^{test}), F(\kappa_j^i))\}) \quad (10)$$

where $C(\kappa)$ denotes the label of class with which a keypoint κ is involved.

Using the class-label noted by each keypoint, we can obtain the class-label of the test image through voting:

$$C(I_{test}) = \arg \max_i \sum_{k=1}^{N_{test}} \delta(i, C(\kappa_k^{test})), \quad (11)$$

where $\delta(\cdot, \cdot)$ denotes the kronecker delta, and N_{test} denotes the number of keypoint in test image.

The proposed voting method is appropriate for facial data with diverse variations. Since important facial features such as eyes, nose, and mouth can vary independently, a keypoint giving high matching score in an image can give low matching score in another image even though the two images are given from the same person. Because the proposed method assigns each keypoint to a class first, it can find a training image with most similar variation at the corresponding local point. Through voting the assignment result of each keypoint, we can find a specific person who has the largest number of most similar features. However, the proposed method does not consider the location of features and the locational mis-match can still occur.

3.3 Hybrid Method with a Threshold

To overcome the drawbacks of the above two methods, we propose a hybrid matching strategy. The proposed matching strategy consists of three steps. In first step, we conduct SIFT on all training images to obtain keypoints. Next, like SIFT-Grid method, each training image is subdivided into different sub-images using a regular grid with overlapping, and we conduct a keypoint-pool for each sub-image as follows:

$$Keypoint-pool = \{\kappa_j^{it} | i = 1, \dots, N, t = 1, \dots, T, j = 1, \dots, N_{it}\} \quad (12)$$

where N represents the number of training images, N_{it} represents the number of keypoints for t th sub-image of i th training image, and T represents the number of sub-images in each training image. The minimum pair distance between a keypoint from a sub-image of test image and the keypoint-pool is computed as follows:

$$D(\kappa_k^{test_t}, Keypoint-pool) = \min_{i,j} (d(F(\kappa_k^{test_t}), F(\kappa_j^{it}))) \quad (13)$$

where $\kappa_k^{test_t}$ represents k th keypoint in t th subregion of the test image. Based on the distance, the class-label of each keypoint, $C(\kappa)$, and the class-label of the test image, $C(I_{test})$ can be found by using the same voting strategies given in Section 3.2. The proposed hybrid matching strategy can give locational information by using regular grid of face image and also utilize the local independent property of keypoints.

Furthermore, when face image is occluded by sun glasses or scarf, it is possible to modify the matching strategy by using threshold technique. Since the obtained keypoints from sun glasses or scarf, should not be used to distinguish between faces, we need to discard them using threshold (See Fig 1). It can be achieved by just taking a keypoint which has smaller minimum pair distance than a threshold as a vote. In next section, we will describe the results of each matching strategy for face recognition.

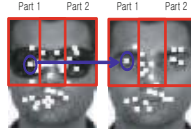


Fig. 1. The sample of occlusion keypoint

4 Experimental Results

In order to verify robustness of the proposed method, we conducted facial classification task on the AR face database [13] which is one of famous benchmark data with various transformations including occlusions. This database contains over 4,000 color images of the frontal faces of 126 subjects. For each subject, there are 26 different images, which were recorded in two different sessions separated by two weeks, each session consisting of 13 images. Images from one subject have various transforms such as expressions, illumination conditions, and occlusions (See Figure 2). All images are of 768×576 pixels and of 24 bits of depth. From this database, we randomly selected 40 different individuals (20 males and 20 females) and used the first-session data set. The original color images with 768×576 pixels were morphed to 85×60 pixel arrays, as stated in [4]. The selected images are cropped by [4] and we converted them to gray-level images. For computational efficiency, the size of the images was reduces to 68×50 pixels. The sample images used in the experiments are shown in Figure 2.

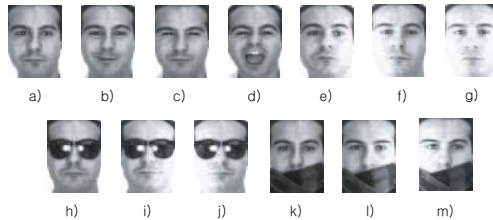


Fig. 2. Images of one subject in the AR face database

Using these images, we constructed two data sets which consist of different images:

– Data set 1

For training, three non-occluded images from each subject (e.g. Figure 2(a), (c), and (g)) are used.

For test, remained four non-occluded images from each subject (e.g. Figure 2(b), (d), (e), and (f)) are used.

– Data set 2

For training, three non-occluded images from each subject (e.g. Figure 2(a), (c), and (g)) are used.

For test, remained four non-occluded images and six occluded images from each subject (e.g. all the images of Figure 2 excluding (a), (c), and (g)) are used.

There are 40 classes and only three units of samples in each class, which may cause instability of LDA. In order to solve the problem arising from a small sample set, we applied PCA prior to using LDA [6].

Table 1. Results of face classification using the AR database

AR Face Database	Data Set 1	Data Set 2
PCA	69.37% _(100dim)	43.25% _(100dim)
LDA	96.88% _(37dim)	60% _(39dim)
SIFT-Grid	75%	64%
Keypoint Voting	84.38%	74%
SIFT-Grid + Keypoint Voting	86.88%	72%
Keypoint Voting _(threshold=0.3)	84.38%	81%
SIFT-Grid + Keypoint Voting _(threshold=0.3)	88.75%	84.25%

The experimental results are shown in Table 1 for each data set. For the both sets, PCA has generally low performances. Therefore, we can conclude that PCA is not robust to various transformations. Unlike PCA, LDA gives best performances than the others in data set 1. However, LDA failed to find meaningful features to distinguish between faces. This means that LDA is not suitable to occlusion data. Although, the performance of the proposed hybrid method is not best in data set 1, it is generally high. In addition, it gives best performance in data set 2. Therefore, we can conclude that the proposed hybrid method is robust to the transformation of occlusion.

5 Conclusions

In order to apply SIFT method for representing facial image data, this paper proposed a hybrid matching strategy which combines the SIFT-Grid method and the keypoint voting with a threshold. The proposed matching strategy consists

of three steps. In first step, we conduct SIFT on all training images to obtain keypoints. Next, each training image is subdivided into different sub-images using a regular grid with overlapping, and we construct a keypoint-pool from sub-images of all training image. Finally, based on the minimum pair distance, each keypoint is assigned to a class first and the class-label of the test image is determined by using the keypoint voting strategy. Therefore, the proposed hybrid matching strategy can give locational information of features by using regular grid of face image, and can also utilize the local independent property of keypoints. In addition, when face image is occluded by sun glasses or scarf, the proposed method can discard occluded keypoints by using a threshold. Through the computational experiments on AR face database, we confirmed the robust performance of the proposed method.

Acknowledgments. This work was supported by National Research Foundation of Korea Grant funded by the Korean Government (2009-0082262.)

References

1. Murase, H., Nayar, S.: Learning Object Models from Appearance. In: AAAI 1993 Proceedings, pp. 836–843 (1993)
2. Turk, M.A., Pentland, A.P.: Face Recognition Using Eigenfaces. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–591 (1991)
3. Mardia, K.V., Kent, J.T., Bibby, J.M.: Multivariate Analysis. Academic Press, London (1979)
4. Martínez, A.M., Kak, A.C.: PCA versus LDA. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23(2), 228–233 (2001)
5. Lee, O., Park, H., Choi, S.: PCA vs. ICA for Face Recognition. In: The 2000 International Technical Conference on Circuits/Systems, Computers, and Communications, pp. 873–876 (2000)
6. Bellhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Trans. on Pattern Recognition and Machine Intelligence* 19(7), 711–720 (1997)
7. Fisher, R.A.: The Statistical Utilization of Multiple Measurements. *Annals of Eugenics* 8, 376–386 (1938)
8. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, London (1990)
9. Lowe, D.G.: Object Recognition from Local Scale-Invariant Features. In: The Proc. of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157 (1999)
10. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60(2), 91–110 (2004)
11. Bicego, M., Lagorio, A., Grosso, E., Tistarelli, M.: On the use of SIFT features for face authentication. In: Computer Vision and Pattern Recognition Workshop, p. 35 (2006)
12. Martínez, A.M., Kak, A.C.: PCA versus LDA. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23(2), 228–233 (2001)
13. Martínez, A.M., Benavente, R.: The AR-face database, CVC Technical Report # 24 (1998)

Selecting, Optimizing and Fusing ‘Salient’ Gabor Features for Facial Expression Recognition

Ligang Zhang and Dian Tjondronegoro

Faculty of Science and Technology, Queensland University of Technology
2 George Street, Brisbane, 4000, Australia
ligzhang@gmail.com, dian@qut.edu.au

Abstract. This paper describes a novel framework for facial expression recognition from still images by selecting, optimizing and fusing ‘salient’ Gabor feature layers to recognize six universal facial expressions using the K nearest neighbor classifier. The recognition comparisons with all layer approach using JAFFE and Cohn-Kanade (CK) databases confirm that using ‘salient’ Gabor feature layers with optimized sizes can achieve better recognition performance and dramatically reduce computational time. Moreover, comparisons with the state of the art performances demonstrate the effectiveness of our approach.

Keywords: Facial expression recognition, Gabor filter, (2D)²PCA, KNN.

1 Introduction

Facial expression recognition (FER) is an active area and has been increasingly given much attention in recent years due to its potential to be applied into a wide range of areas, including human-computer interaction, video surveillance, video indexing and summarization. To date, a robust FER is still a challenging issue due to facial image variations, such as illumination, rotation and occlusion.

FER method can be classified into four categories: motion-based, feature-based, model-based and appearance-based approaches. Appearance-based is the most effective approach to handle facial image in real situations since it is insensitive to in-plane rotation and illumination variations, particularly Gabor filter. However, there are three weaknesses in the use of Gabor filter which need to be overcome, including redundant information within the neighboring frequencies [1]; expensive computation [2]; and different channels have different contributions on recognition performance [3]. In this paper, we will address these problems by selecting ‘salient’ Gabor filters. There have only been few studies on the ‘salient’ Gabor features selection, which can be categorized into three groups: 1) Point based approach [4, 5] which extracts Gabor features based on fiducial points of a face grid. However, its recognition performance is dependent on the accuracy of the automatically selected and located fiducial points, which is still a challenging task. 2) Feature based approach which performs Gabor filters on facial images and selects the ‘salient’ features using feature selection algorithms such as Adaboost [6, 7], genetic programming (GP) [8] and zero norm [9]. Although it overcomes the drawback of point based approach, it still requires accurate face location. 3)

Channel based approach [3] which aims to select a subset of Gabor channels corresponding to different scales and orientations. Unlike the other two methods, this approach eliminates the requirement of point location at the cost of losing expressional information in unselected channels. The selection can be specifically optimized for each expression [10] or overall performance [1].

In this paper, we propose a channel based approach that selects, optimizes and fuses a set of ‘salient’ Gabor filters for effective FER from still images. We extend Gabor filters from 5 to 18 scales and adopt $(2D)^2$ PCA instead of PCA for dimension reduction. The selection of feature layers and the determination of their optimized sizes are automatically processed based on the recognition performance of an image set. The selected ‘salient’ layers are fused for six universal expressions recognition, including anger AN, disgust DI, fear FE, happy HA, sadness SA and surprise SU, using the K nearest neighbor (KNN) classifier.

The main contributions are as follows: 1) We propose a novel and automatic approach to select and optimize ‘salient’ Gabor features. To the best of our knowledge, our approach is the first attempt to exploit the selection of ‘salient’ Gabor features from the aspect of scale, orientation and size. Meanwhile, our approach also is the first one to explore the way of determining the optimized sizes of feature matrixes. 2) We investigate the recognition performances of KNN using K values ranged from 1 to 14. Our results indicate that the best performance is obtained when K equals to 1. 3) We use $(2D)^2$ PCA for dimension reduction. Our results show that it only takes a small proportion of the overall computational time. 4) We confirm that using ‘salient’ features can lead to a better performance with dramatically less computational time than using all features. 5) We present results to confirm Littlewort’s finding [7] that useful emotional features are distributed in a wide range of Gabor feature scales. 6) We use a comprehensive evaluation to demonstrate that “sad” contributes to most of the misrecognitions, while “surprise” is the easiest facial expression to be correctly recognized for both JAFFE and CK databases.

The rest of the paper is organized as follows. Section 2 describes in details the proposed framework and each step. Section 3 shows the performance evaluations using three types of comparisons, namely approach using all features, computational time and state of the art performances. Finally, conclusions are drawn in section 4.

2 System Framework

The proposed framework as shown in Fig. 1 is composed of five steps: pre-processing, Gabor features, $(2D)^2$ PCA, layer selection and layer fusion. During pre-processing, face images are cropped and scaled into a resolution of 110*110 pixels. These images are then passed through 9 bands, 2 scales, and 4 orientations Gabor filters. In this paper, we define a *layer* as a Gabor feature representation with different bands, scales and orientations. These layers are processed by $(2D)^2$ PCA for dimension reduction, which produces feature matrix layers with the same bands, scales and orientations, but smaller sizes. Layer selection is then automatically achieved based on the performance of an image set to choose the most ‘salient’ feature matrix layers and decide their optimized sizes. Finally, the ‘salient’ optimized layers are fused for recognizing the six universal expressions using the KNN classifier.

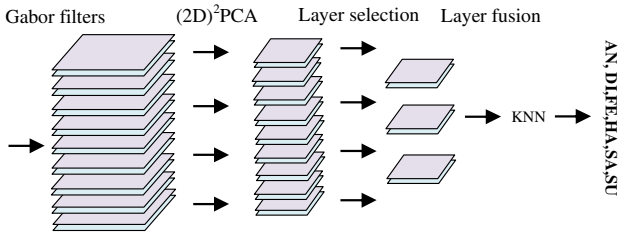


Fig. 1. Flow chart of the proposed framework

2.1 Gabor Features

Gabor filters have been successfully applied to a wide range of fields, such as face recognition [11] and fingerprint identification [12]. In this paper, 2D Gabor filter is adopted and it can be mathematically expressed as:

$$F(x, y) = \exp\left(-\frac{(X^2 + \gamma^2 Y^2)}{2\sigma^2}\right) \times \cos\left(\frac{2\pi}{\lambda} X\right) \cdot$$

$$X = x \cos \theta + y \sin \theta \quad Y = -x \sin \theta + y \cos \theta \quad (1)$$

where, orientation θ , the effective width σ , the wavelength λ , the aspect ratio $\gamma = 0.3$. In this paper, 9 bands, 2 scales in each band, and 4 orientations ($90^\circ, -45^\circ, 0^\circ, 45^\circ$) are adopted. The values of these parameters are set based on [13]. Given an image, each pixel is convoluted with Gabor filters, resulting in a series of Gabor images with expressional features (e.g. bar and edge).

2.2 (2D)²PCA

PCA-based methods have been widely used for dimension reduction, however, most of the methods need to reshape 2D image into a 1D feature vector, which leads to three problems: the intrinsic 2D structure of an image is removed, curse of dimensionality dilemma and small sample size [14]. Thus, (2D)²PCA [15] was used in our framework to directly calculate the feature without matrix-to-vector conversion, and save storage requirement by performing PCA on row and column pixels simultaneously to obtain feature matrixes that represent images.

2.3 Layer Selection

The tasks of selecting ‘salient’ matrix layers and determining their optimized sizes are completed by using the recognition performance of an image set from the JAFFE database. The test set includes images with the emotion index ‘1’, whilst the training set comprises of the rest images. As for optimized sizes, a size range [4, 40] with an interval of 2 is chosen based on preliminary experiments. The selection process can be described as follows.

Let L_{bsot} be the b^{th} band, s^{th} scale and o^{th} orientation layer of training image A_t ($b = 1, 2, \dots, 9; s = 1, 2; o = 1, 2, 3, 4; t = 1, 2, \dots, M$, M is the number of training images), the

feature matrix of L_{bsot} is F_{bsot} . Let L_{bsot} be the b^{th} band, s^{th} scale and o^{th} orientation layer of test image T_l ($l = 1, 2, \dots, Q$; Q is the number of test images), the feature matrix of L_{bsot} is F_{bsot} . The distance between L_{bsot} and L_{bsot} is defined by

$$D(L_{bsot}, L_{bsot}) = \|F_{bsot} - F_{bsot}\| \tag{2}$$

where $\|\cdot\|$ is the L1 or L2 norm of $(F_{bsot} - F_{bsot})$.

Then, the correct recognition rate (CRR) of the b^{th} band, s^{th} scale and o^{th} orientation layer of all test images can be obtained by the nearest neighbor classifier using these distances. Based on the results, layers with comparatively higher CRRs are selected as ‘salient’ layers. For each ‘salient’ layer, the optimized size is set to be a little bigger than the size of the best performance in order to gain a general performance. Finally, a total of 26 ‘salient’ layers and their optimized sizes are obtained and listed in Table 1, in which BSO ‘322’ represents the 3th band, 2th scale and 2th orientation, L2 stands for using L2 distance.

Table 1. The selected ‘salient’ feature matrix layers and sizes

BSO	Size	BSO	Size	BSO	Size	BSO	Size	BSO	Size
322	20	513	22	622(L2)	18	723	16	913	16
412	20	522	16	623	10	724	14	923	14
413	20	523	14	624	16	812	10	-	-
422	18	612(L2)	18	712	12	813	16	-	-
423	18	613	20	713	18	814	22	-	-
512	18	614	16	714	14	823	14	-	-

2.4 Layer Fusion

The layer fusion step performs FER by fusing the ‘salient’ feature matrix layers with optimized sizes. Firstly, for each ‘salient’ layer of one test image, KNN is used to calculate the K possible expressions. Then the expressions of all layers are combined to obtain the final result using the maximum rule. The algorithm is as follows.

Let L_{pt} be the p^{th} layer of training image A_t ($p = 1, 2, \dots, 26$; $t = 1, 2, \dots, M$), and L_{pl} be the p^{th} ‘salient’ layer of test image T_l ($l = 1, 2, \dots, Q$), their feature matrices are F_{pt} and F_{pl} respectively. For each L_{pt} , the M distances $D(L_{pt}, L_{pl})$ between L_{pt} and L_{pl} of all training images can be calculated by the equation (2). The nearest distance of the M distances is defined by

$$D(L_{pt}, L_{pl}) = \min_{i=1}^M \|F_{pt} - F_{pl}\| \tag{3}$$

Similarly, the K smallest $D(L_{pt}, L_{pl})$ also can be obtained, the emotion labels E_{pi}^g ($i = 1, 2, \dots, K$; $g = 1, 2, \dots, 6$; $E_{pi}^g \in \{AN, DI, FE, HA, SA, SU\}$) of these chosen K L_{pk} are recorded. Then E_{pi} with the same emotion label are summed over 26 ‘salient’ layers:

$$E_g = \max_{g=1}^6 \left(\sum_{p=1}^{26} \sum_{i=1}^k E_{pi}^g \right) \tag{4}$$

Thus, the final output of emotion g corresponds to the largest E_g .

3 Experiments

3.1 Databases

The JAFFE database [4] contains 213 gray images of 7 facial expressions posed by 10 Japanese females. Each object has 3 or 4 frontal face images for each expression. The name of each image is identified by subject name initials, emotion initials & index, and image index. Cohn-Kanade database [16] includes 2105 image sequences from 182 subjects ranged in age from 18 to 30 years. Image sequences were digitized from neutral to target display. The six universal expressions were based on descriptions of prototypic emotions. In this paper, all the images of the six universal expressions from JAFFE are used. For CK, 1184 images that represent one of the six expressions are selected, 4 images for each expression of totally 92 subjects. The images are chosen from the last image of each sequence, then one every two images. The faces of all images from two databases are cropped and scaled to a resolution of 110*110 pixels.

3.2 JAFFE Database Tests

For each validation step, the images with the same emotion index are grouped as the test set, and the remaining images are regarded as the training set. In this research, only emotion index from 1 to 3 are tested due to the fact that most subjects do not contain images with emotion index '4'. As a test benchmark, all layer (AL) approach is defined as using all layer features and L1 distance.

The CRRs of three test sets using KNN with K ranged from 1 to 14 are shown in Fig. 2. As shown in this figure, for both the proposed and AL approaches, the highest CRR of each set is obtained by KNN when K=1. Regarding the highest CRR of each set, the proposed approach shares the same value (90.0%) with AL approach in set1 and achieves bigger values than all layer approach in set2 and set3. The highest CRR (96.923%) of the proposed approach is 3.077% bigger than that of the AL approach. Therefore, it can be concluded that the chosen and optimized layers can achieve better recognition performance than using all layers. Since the training and test images of set2 and set3 are different from those used for obtaining the chosen and optimized layers, their high performances indicate a good general recognition capability of these 'salient' layers.

Among the three sets, set2 obtains the best overall recognition performance for all K values and keeps the highest CRRs in both the proposed and AL approaches, while set1 ranks the lowest. After the peak performance in both approaches, the CRRs decrease as K values increase, whereas CRRs of the proposed approach decrease quicker than those of AL approach. The reason is probably that AL approach utilizes all expressional information for FER, thus a steady decline of CRR is expected, whereas the proposed approach only adopts part of this information, therefore, a rapid decrease is anticipated.

The confusion matrix of the six expressions can be drawn by setting K to be 1 for all three sets in order to obtain the highest CRRs. The result is demonstrated in Table 2. As shown in this table, the images of surprise are all correctly recognized probably due to the apparent characteristic of big mouth; the second best recognized emotion is

anger, and only one image is falsely classified as sad. On the other hand, happy is the most difficult emotion to distinguish from others. Another interesting point is that sad is the emotion that is most likely to be incorrectly recognized as target emotion. And this may be owing to the erratic expressers on sad in JAFFE, which is in accord with the work [5] that reported two erratic expressers (UY and NA) existed in JAFFE.

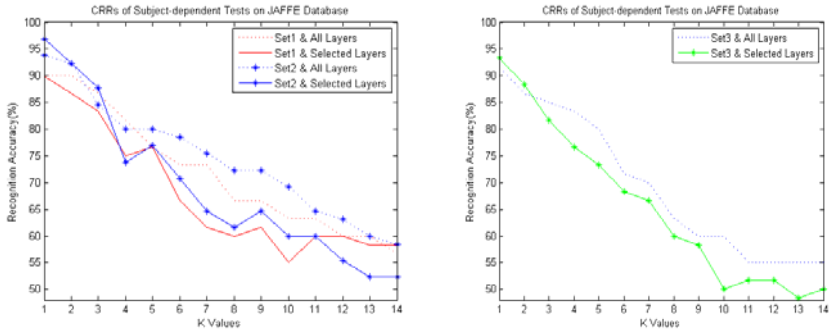


Fig. 2. CRR comparisons between the proposed and AL approaches using JAFFE database

Table 2. Confusion matrix of six expressions using JAFFE database

	AN	DI	FE	HA	SA	SU	Overall
AN	29	0	0	0	1	0	96.7%
DI	0	28	1	0	0	1	93.3%
FE	0	1	27	0	2	0	90.0%
HA	0	0	0	26	4	0	86.7%
SA	0	0	1	1	28	0	93.3%
SU	0	0	0	0	0	30	100%

3.3 CK Database Tests

Since each subject has four images for each expression, all images can be classified into four sets that include one of the four images per set. Four cross-validation tests are conducted separately and the results are compared with the AL approach as shown in Fig. 3. Based on the graphs, the overall performances of the two approaches are fairly satisfactory. For all the four sets, both approaches achieve their highest CRRs when K=1 and 2, but the AL approach can retain the highest CRR of 100% with a big K value (for instance, 6 in set3). As for set1, set2 and set3, the highest CRRs of both approaches is 100%, while for set4, the highest CRR (99.662%) of the proposed approach is 0.338% lower than that of the AL approach since one happy image is wrongly recognized as fear. For both approaches, the CRRs decrease when K increases. However, similar to our findings while using JAFFE database, CRR of the proposed approach declines quicker than that of the AL approach. Thus, we can conclude that the selected ‘salient’ layers can achieve higher recognition performances compared to using all layers.

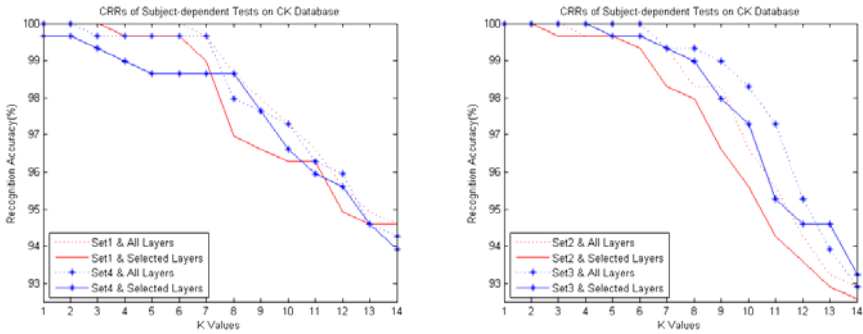


Fig. 3. CRR comparisons between the proposed and AL approaches using CK database

3.4 Computational Time Comparison

For each of JAFFE and CK, the average computational time of all test images at three stages, including Gabor feature, $(2D)^2$ PCA and KNN, is calculated and demonstrated in Table 3. The program was developed by Matlab 7.0.1 under a laptop configuration of core duo 1.66GHz CUP and 2GB memory. Based on the time, the proposed approach has shown a substantial improvement compared with the AL approach as it has reduced 75% to 80% of the processing time in the AL approach. Moreover, there is an 75% to 82% of time reduction for computing Gabor features, 65% to 75% for processing $(2D)^2$ PCA, and 75% to 80% for recognizing expressions using KNN. Time spent on computing Gabor features is nearly 90% of the overall time for JAFFE, and about 60% for CK. This demonstrates that Gabor feature is the most computationally expensive. On the other hand, $(2D)^2$ PCA only requires 2.7% to 4.7% of the overall time. Another notable point is that the computational time of KNN on CK is 6 to 7 times as much as that on JAFFE. This is due to KNN has a bigger number of test and training images to process as CK contains more images than JAFFE.

Table 3. Computational time comparisons at three stages (in seconds)

	Proposed approach				All layer approach			
	Gabor	$(2D)^2$ PCA	KNN	Total	Gabor	$(2D)^2$ PCA	KNN	Total
JAFFE	0.301	0.016	0.025	0.342	1.263	0.047	0.101	1.411
CK	0.244	0.011	0.150	0.405	1.342	0.047	0.711	2.100

3.5 Comparisons with Previous Work

In this paper, the performances of Liang [17] (using LLE) and Guo [18] (using FSLP) are used as the benchmark for JAFFE, while the performances of Wang [19] (using NBC and QDC) and Wong [20] (using FEETS) are used as the benchmark for CK. The choice on these benchmarked works is based on the database images being the most similar to our work. The comparison results are shown in Table 4, from which we can see that using JAFFE, the proposed approach exceeds Liang’s approach by 1.90% with respect to the maximum (Max) CRR, 0.6% for the average (Ave) CRR, and 4.3% for the minimum (Min) CRR. Moreover, it exceeds Guo’s approach by

2.4% for the Ave CRR. A better performance is shown by the CK database as the proposed approach surpasses Wong’s by 6.71% for the Max CRR and 17.14% for the Min CRR, while it also surpasses Wang’s approach by 3.43% for the Max CRR, and 12.45% for the Min CRR. Hence, our experiment has demonstrated a significant recognition improvement in the proposed approach compared to the previous work.

Table 4. CRR comparisons with previous work (%)

	Proposed approach			[17] and [19]			[18] and [20]		
	Max	Ave	Min	Max	Ave	Min	Max	Ave	Min
JAFFE	96.9	93.4	90.0	95	92.8	85.7	-	91.0	-
CK	100	99.89	99.66	93.29	-	82.52	96.57	-	87.21

4 Conclusions

This paper presents a novel method to automatically select, optimize and fuse ‘salient’ Gabor layers to improve the current performance in FER from still images. The experiments on JAFFE and CK databases demonstrate that the proposed approach can achieve significant improvements on recognition performance and computational time compared to the previous work. Our results confirm that wider range of Gabor filters can improve the performance as expressional information is evenly distributed over these filters. Moreover, our experiments show that the time used for computing Gabor filters takes a large part of the overall processing time of our framework, while (2D)²PCA only requires a small proportion of the overall time.

In our future work, we aim to conduct more experiments to improve CRR by increasing orientation number. Meanwhile, the combination of (2D)²PCA with other local feature extraction methods (for example, local binary pattern [21]) seems to be a promising direction. Another important field is combining both appearance and motion features for FRE since researches [22] have confirmed the significant role of dynamic information in the process of expressing and recognizing facial expressions.

Acknowledgments. The authors would like to thank Nicki Ridgeway for providing the Cohn-Kanade AU-Coded Facial Expression Database and the providers of JAFFE database.

References

1. Deng, H.B., Jin, L.W., Zhen, L.X., Huang, J.C.: A new facial expression recognition method based on local gabor filter bank and pca plus lda. *International Journal of Information Technology* 11, 86–96 (2005)
2. Caifeng, S., Shaogang, G., McOwan, P.W.: Robust facial expression recognition using local binary patterns. In: *IEEE International Conference on Image Processing, ICIP 2005*, vol. 2, pp. II-370–II373 (2005)
3. Wei Feng, L., ZengFu, W.: Facial Expression Recognition Based on Fusion of Multiple Gabor Features. In: *18th International Conference on Pattern Recognition, ICPR 2006*, vol. 3, pp. 536–539 (2006)

4. Lyons, M., Akamatsu, S., Kamachi, M., Gyoba, J.: Coding facial expressions with Gabor wavelets. In: Proceedings of Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998, pp. 200–205 (1998)
5. Bashyal, S., Venayagamoorthy, G.K.: Recognition of facial expressions using Gabor wavelets and learning vector quantization. *Engineering Applications of Artificial Intelligence* 21, 1056–1064 (2008)
6. Chen, H.Y., Huang, C.L., Fu, C.M.: Hybrid-boost learning for multi-pose face detection and facial expression recognition. *Pattern Recognition* 41, 1173–1185 (2008)
7. Littlewort, G., Bartlett, M.S., Fasel, I., Susskind, J., Movellan, J.: Dynamics of facial expression extracted automatically from video. *Image and Vision Computing* 24, 615–625 (2006)
8. Yu, J., Bhanu, B.: Evolutionary feature synthesis for facial expression recognition. *Pattern Recognition Letters* 27, 1289–1298 (2006)
9. Gunes, T., Polat, E.: Feature selection for multi-SVM classifiers in facial expression classification. In: 23rd International Symposium on Computer and Information Sciences, ISICIS 2008, pp. 1–5 (2008)
10. Lajevardi, S.M., Lech, M.: Facial Expression Recognition Using Neural Networks and Log-Gabor Filters. In: Digital Image Computing: Techniques and Applications, DICTA 2008, pp. 77–83 (2008)
11. Kong, A.: An evaluation of Gabor orientation as a feature for face recognition. In: 19th International Conference on Pattern Recognition, ICPR 2008, pp. 1–4 (2008)
12. Dadgostar, M., Tabrizi, P.R., Fatemizadeh, E., Soltanian-Zadeh, H.: Feature Extraction Using Gabor-Filter and Recursive Fisher Linear Discriminant with Application in Fingerprint Identification. In: Seventh International Conference on Advances in Pattern Recognition, ICAPR 2009, pp. 217–220 (2009)
13. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust Object Recognition with Cortex-Like Mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 411–426 (2007)
14. Kong, H., Wang, L., Teoh, E.K., Li, X., Wang, J.-G., Venkateswarlu, R.: Generalized 2D principal component analysis for face image representation and recognition. *Neural Networks* 18, 585–594 (2005)
15. Zhang, D., Zhou, Z.-H. (2D)²PCA: Two-directional two-dimensional PCA for efficient face representation and recognition. *Neurocomputing* 69, 224–231 (2005)
16. Kanade, T., Cohn, J.F., Yingli, T.: Comprehensive database for facial expression analysis. In: Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 46–53 (2000)
17. Liang, D., Yang, J., Zheng, Z., Chang, Y.: A facial expression recognition system based on supervised locally linear embedding. *Pattern Recognition Letters* 26, 2374–2389 (2005)
18. Guo, G., Dyer, C.R.: Learning from examples in the small sample case: face expression recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 35, 477–488 (2005)
19. Wang, J., Yin, L.: Static topographic modeling for facial expression recognition and analysis. *Comput. Vis. Image Underst.* 108, 19–34
20. Wong, J.-J., Cho, S.-Y.: A face emotion tree structure representation with probabilistic recursive neural network modeling. *Neural Computing & Applications* (2008)
21. Shan, C., Gong, S., McOwan, P.W.: Facial expression recognition based on Local Binary Patterns: A comprehensive study. *Image and Vision Computing* 27, 803–816 (2009)
22. Yongmian, Z., Qiang, J., Zhiwei, Z., Beifang, Y.: Dynamic Facial Expression Analysis and Synthesis With MPEG-4 Facial Animation Parameters. *IEEE Transactions on Circuits and Systems for Video Technology* 18, 1383–1396 (2008)

Self-Organized Gabor Features for Pose Invariant Face Recognition

Saleh Aly¹, Naoyuki Tsuruta², and Rin-ichiro Taniguchi¹

¹ Department of Intelligent Systems, Kyushu University,
744 Motoooka, Nishi-Ku, Fukuoka, 819-0395, Japan
{aly,rin}@limu.ait.kyushu-u.ac.jp

² Department of Electronics Engineering and Computer Science, Fukuoka University
8-19-1, Nanakuma, Jonan, Fukuoka 814-0180, Japan
tsuruta@tl.fukuoka-u.ac.jp

Abstract. Pose-invariant face recognition using single frontal training image is considered one of the most difficult challenges in face recognition. To address this problem, we introduce a novel feature extraction method based on learning the manifold of local features. Changes in local features due to pose variations induce a nonlinear manifold in the feature space. Self-organizing map is employed to learn the manifold induced by Gabor filter response of a generic training face database captured at various pose directions. Furthermore, this manifold can be used to represent new face image as a set of points in the feature space. A modular Hausdorff distance measure, which can effectively measure the similarity between two point sets without any correspondence, is also proposed to identify unlabeled subjects. Experimental results on CMU-PIE face database show the effectiveness of the novel method against pose variations.

1 Introduction

Although humans can detect and identify faces in a scene without much effort, building an automated system that accomplishes such tasks is very challenging [1]. The challenges are even more profound when one considers the wide variations in imaging conditions [2]. There are inter- and intra-subject variations associated with face images. Inter-subject variation is limited due to the physical similarity among individuals. Intra-subject variation, on the other hand, is very extensive and can be attributed to three factors: pose, illumination, and expression. In this paper, we are concerned with building a robust face recognition system against pose variation.

Pose invariant algorithms can be broadly classified into three categories, invariant feature-based, 2D view-based, and 3D-based techniques. The invariant feature-based approach records expressive features in a face image that do not vary under pose changes. These methods can be divided into appearance-based like Eigenfaces [3] and Fisherfaces [4] algorithms, geometric model-based algorithms like Elastic Bunch Graph Matching [5] and Active Appearance Model

[6]. Although these approaches are simple and fast, their performance is sensitive to misalignment and can not separate image variance caused by identity and pose variations. The 2D view-based approach stores a set of observed multiview images in the gallery to deal with pose variation problem [7] or synthesized new view images from a given image [8]. Moreover, invariance can be achieved by pose transformation in the image space [9] or pose transformation in the feature space [10]. However, linear transformation cannot adequately describe image variations caused by pose changes. In the third category, 3D generic face model [11][12] assist to predict the appearance of a face under different pose parameters. However they are computationally expensive and require manual locating of facial features. In this paper, we present a new algorithm belongs to the first category, however the proposed algorithm does not require any prior facial alignment or any feature localization.

A self-organized Gabor feature (SOGF) method is introduced to nonlinearly model the extracted local facial features. In order to distinguish between local features extracted from different facial regions, input image is divided into three horizontal regions and three feature maps are learned from each major facial part (i.e. eyes, nose, and mouth). SOGF is used as a feature extractor to represent facial image parts in a new invariant feature space by learning such variations from generic training data. In order to minimize the redundancy between features, self-organizing map (SOM) is employed to transform the high dimensional Gabor feature space into nonlinear low dimensional topological space, which capture the intrinsic dimensionality of Gabor feature space. Finally, modular Hausdorff distance measure, which can effectively exploit the output of the SOM topological space, is also proposed to identify unlabeled subjects.

This paper is organized as follows. In Section 2, we discuss the framework of self-organized Gabor feature based face recognition system. More details about each component of the system is given in the following subsections. Experimental results are presented in Section 3. Finally, the summary and conclusions are given in Section 4.

2 Proposed Face Recognition System

In this paper, we propose a novel self-organized Gabor features (SOGF) method for face recognition whose system architecture is shown in Fig. 1. Gabor filters are used as a local feature extractor for the input face image and SOM is used to nonlinearly project Gabor features and represent them in a new feature space learned from pose-variant training data. The projected face region called *component-map*. The proposed system contains three component-maps, upper component-map represents features extracted from eye region, middle component-map represents features derived from nose, and lower component-map represents features from mouth region. Features extracted by each map are used to represent variations in local features of the input face region. The rationale behind integrating Gabor wavelet and SOM is two-fold. On the one hand, the Gabor transformed face images exhibit strong characteristics of spatial locality, scale and orientation selectivity. These images can thus produce salient local

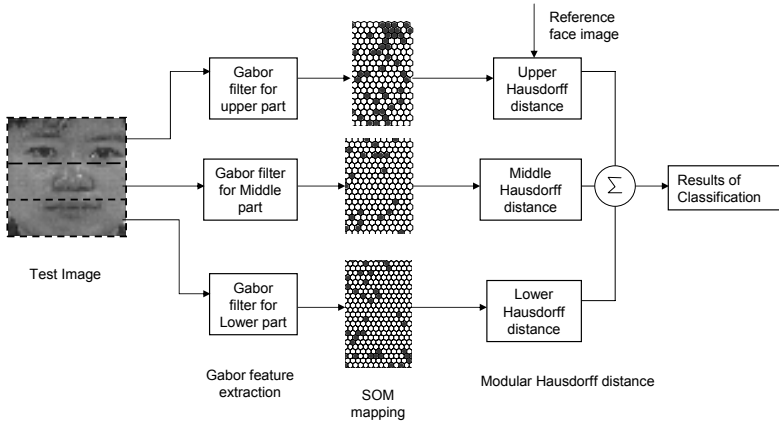


Fig. 1. Architecture of proposed face recognition system

features that are most suitable for face recognition. On the other hand, SOM can model the nonlinear manifold of the local features caused by pose variations. Furthermore, SOM would further reduce the redundancy and represent Gabor features in a set of topological ordered nodes. These features are encoded only by the position of the best matching nodes in SOM maps.

Intuitively, out-of-plane face rotation (face rotation around vertical axis) leads to disappearance of some facial parts or shrinking them into smaller area. For example, part of the eyes, nose and mouth regions are occluded due to viewpoint changes. Since representing all local features by one global face-map will destroy the configuration and the discrimination characteristics of the three main facial parts (eyes, nose, and mouth). Horizontal partitioning of the face image into three regions is an appropriate choice to retain distinctive information of the major feature points against pose variations. The resulting Gabor response image is divided into three horizontal parts, upper, middle and lower part. Feature matching between two feature maps is carried out without any correspondence using Hausdorff distance. We will give more details about each component of the system in the following sections.

2.1 Gabor Wavelet

Gabor filters are commonly used for image analysis because of their biological relevance and computational properties. The Gabor wavelets, whose kernels are similar to the 2D receptive field profiles of the mammalian cortical simple cells, exhibit strong characteristics of spatial locality and orientation selectivity, and are optimally localized in the space and frequency domain. The following form of a 2D Gabor filter function in the continuous spatial domain has been employed:

$$\psi(x, y, \theta, \lambda, \gamma) = e^{-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}} \cos(2\pi \frac{x'}{\lambda}) \tag{1}$$

$$x' = x \cos \theta + y \sin \theta, y' = -x \sin \theta + y \cos \theta \quad (2)$$

where $\lambda, \theta, \gamma, \sigma$ specify the wavelength, orientation, aspect ratio of the wavelet and the radius of Gaussian respectively.

Tuning Gabor wavelets parameter is a very complex task, the filter parameters used in this paper are inspired by the work in [13] because it gives biologically plausible Gabor filters which perform well for filtering tasks related with object recognition. However in our implementation we consider only 8 orientations and 5 scales as recommended by [5]. The filters are arranged to form a pyramid of scales, $\lambda \in \{4, 4\sqrt{2}, 8, 8\sqrt{2}, 16\}$, and these filters span a range of sizes from 9×9 to 17×17 pixels in steps of two pixels. The orientation parameter is sampled into 8 different orientations over the interval from 0 to π , i.e., $\theta \in \{0, \frac{\pi}{8}, \frac{2\pi}{8}, \frac{3\pi}{8}, \frac{4\pi}{8}, \frac{5\pi}{8}, \frac{6\pi}{8}, \frac{7\pi}{8}\}$. The radius of Gaussian function is set such that the wavelets of different size and frequency are scaled versions of each other, i.e., $\sigma = \lambda$. The aspect ratio parameter is included such that the wavelets could also approximate some biological models, and the wavelets used in this paper have circular Gaussian, i.e., $\gamma = 1$. Fig. 2 shows Gabor filter kernels at five scales and eight orientations.

Gabor Feature Vector. Let $I(x, y)$ be the gray level distribution of an image, and the Gabor wavelet representation of image $I(x, y)$ is defined as follows:

$$O(x, y, \theta, \lambda, \gamma) = I(x, y) * \psi(x, y, \theta, \lambda, \gamma) \quad (3)$$

where $*$ denotes the convolution operator. Due to the misalignment between input images, a small shift in the response of filters is produced. In order to compensate this shift variation, a MAX filter is applied to local areas in the Gabor-filtered image [13]. In our experiments the size of the MAX filter is 2×2 pixels. As a result of this operation, the dimension of the output image reduced by half, which consequently reduce the computational cost of the successive operations. Furthermore, the response of each Gabor filter is normalized to zero mean and unit variance.

2.2 Self-Organizing Map

In self-organizing map (SOM) [14], a pattern is projected from an input space to a position in the map where the information is coded as the location of the activated neuron. The SOM is unlike most classification or clustering techniques in that it provides a topological ordering of data. Similarity in the input space is preserved in the output space. The topological preservation of the SOM process makes it especially useful in the classification of data, which includes smooth variations in the data. For example, the changes in the face viewpoint cause distortions in its local features, which can be modeled either by the Voronoi region of the winner neuron in the feature map or even by its neighbor neurons.

The SOM codebook has two important characteristics that make it especially suitable for nonlinear projection. First, the probability density function (PDF) of the codebook is a good approximation for the PDF of the training data.

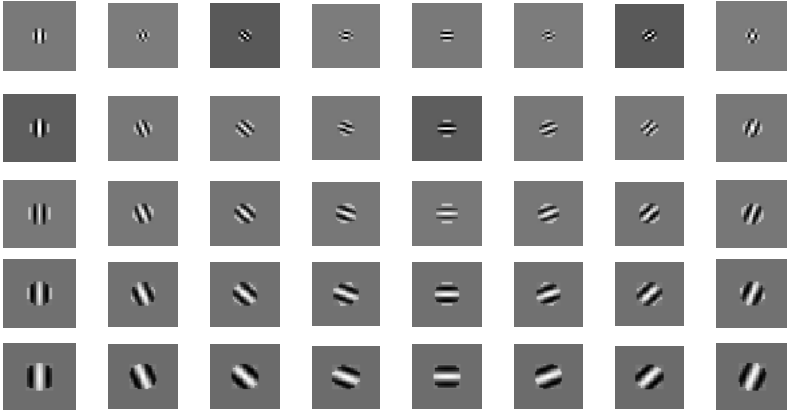


Fig. 2. Gabor wavelets at five scales and eight orientations

Second, the topographic order of the training data is preserved in the codebook, even if the dimensionality of the SOM is smaller than that of training data. The second characteristic means that similar features are mapped to nearby positions in the feature map. However, if the dimensionality of the map is too low, the map tries to approximate the high dimensionality input space by folding itself into the input space. Therefore, a correct dimension of the map leads to better topographic order of neurons.

Local features to be trained with SOM are extracted by convolving each pixel of the input image with a set of Gabor filters. Each neuron in the SOM learns local characteristics of the face image. After training, the feature vector at each pixel is projected in the SOM map and represented by the position of the winner neuron. In our method, features extracted from each region of the face are depressively represented in terms of positional relationship among the winner neurons, each of which corresponds to local feature of the face image. Fig. 3 shows feature maps of the eye region for one person at 5 different pose conditions. This figure indicates that features maps are very similar to each other despite the viewpoint changes in the appearance of the face. Because of the self-occlusion of local feature due to pose variations, the number of active neurons is different for the maps.

2.3 Modular Hausdorff Distance

To identify an unlabeled face, a classifier should be built on the top of SOM maps. Although there are many metrics that can be used to calculate the similarity, we have employed Hausdorff distance, which was previously used for shape matching in face images. Hausdorff distance exhibits tolerance to the translation and distortion variations in the local features caused by pose changes. The Hausdorff distance is defined as a distance between two point sets, and it gives a measure

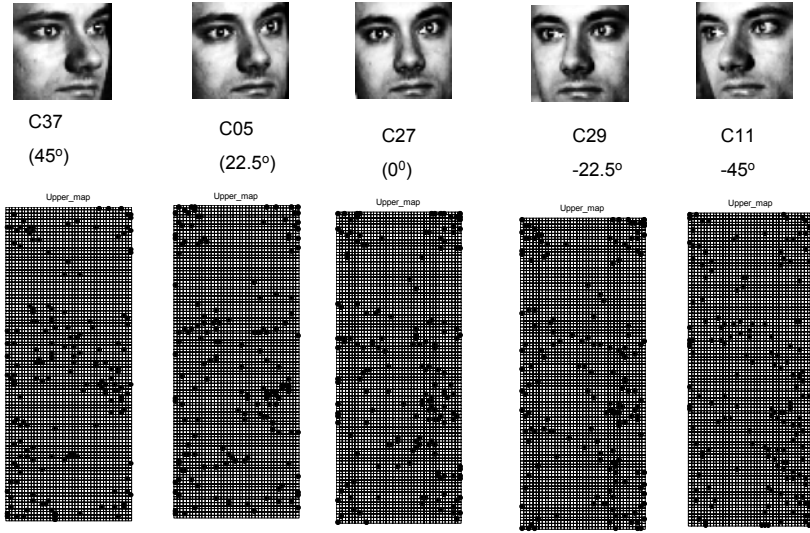


Fig. 3. Eye component-map for one subject from CMU-PIE database under 5 different viewpoints: black dots represent winner neurons

of dissimilarity between the point sets. One advantage of using Hausdorff distance for local feature matching is that it does not require the explicit pairing of points. In [15], the authors revised the metric and investigated 24 different distance measures based on their behavior in the presence of noise, and they redefined the original definition of the metric proposing an improved measure, called the modified Hausdorff distance, which is formulated as:

$$h_p(P, Q) = \frac{1}{N_p} \sum_{p=1}^m \min_{q \in Q} \|p - q\| \tag{4}$$

where $P = \{p_1, p_2, \dots, p_m\}$ and $Q = \{q_1, q_2, \dots, q_n\}$ are two point sets and N_p , represent the cardinality of P set.

In this paper, we adopt the above formulation to calculate the similarity between each part of the face. Three different Hausdorff distance values are obtained from each region of the face and combined to give the final dissimilarity measure. The following equation used to calculate the distance for the whole three face parts.

$$HD(P, Q) = \sum_{p=1}^3 H_p(P, Q), \tag{5}$$

where $H_p(P, Q)$ is the partial directed Hausdorff distance between two face regions and is given by this equation:

$$H_p(P, Q) = \max(h_p(P, Q), h_p(Q, P)) \tag{6}$$



Fig. 4. Pose variations in the CMU-PIE face database. The pose varies from full left profile (c34) to full frontal (c27) and to full right profile (c22).

3 Experimental Results

In the following experiments, we used CMU-PIE database [16], which consists of 68 subjects under 43 significant illumination variation, and with 13 poses and 2 facial expressions. Typical images from one subject for all pose conditions are shown in Fig. 4. All face images under normal light conditions and 7 pose variations are selected. All images in CMU-PIE databases are scaled to the size 48×48 pixels, and photometrically normalized by histogram equalization. We divided the dataset into three disjoint subsets according to the same procedure of [9]

- Generic training data: The generic data are used to construct local feature maps for each face image part.
- Gallery: The gallery is the set of reference images of the people to be recognized. (i.e., the images given to the algorithm as examples of each person who might need to be recognized).
- Probe: The probe set contains the "test" images (i.e., the images to be presented to the system to be classified with the identity of the person in the image).

The division into these subsets is performed as follows: First we randomly select half of the subjects as the generic training data. After the generic training data have been removed, the remainder of the database is divided into probe and gallery sets based on the pose of the images. In All experiments, we set the gallery to be the frontal images c_{27} and the probe set to be $\{c_{37}, c_{05}, c_{09}, c_{07}, c_{29}, c_{11}\}$ poses which span the pose from half left profile ($+45^\circ$) to half right profile (-45°). In this case, we evaluate how well our algorithm is able to recognize people from their profiles when the algorithm has seen them only from the front.

In the first experiment, SOM maps with different sizes are constructed to learn the distribution of local facial features. Upper, middle and lower maps are trained from the patterns at the eyes, nose and mouth regions respectively. All SOMs are trained in batch-mode using all extracted Gabor features and

Table 1. Average recognition rate of SOGF algorithm with varying map dimension

Map Dimension	c_{37}	c_{05}	c_{09}	c_{07}	c_{29}	c_{11}	Mean
[4000]	82	94	94	94	94	73	88
[100 × 40]	94	100	97	100	100	97	98
[40 × 20 × 5]	88	97	94	97	97	88	93
[20 × 10 × 5 × 4]	94	94	88	94	94	91	92
[10 × 8 × 6 × 4 × 2]	97	100	100	100	100	100	99.5
[8 × 6 × 5 × 4 × 2 × 2]	91	100	100	100	100	94	97.5
[8 × 5 × 4 × 3 × 2 × 2 × 2]	100	100	94	100	100	88	97

Table 2. Comparison between Eigenfaces, Fisherfaces, and SOGF algorithm performance across pose

Map Dimension	c_{37}	c_{05}	c_{09}	c_{07}	c_{29}	c_{11}	Mean
Eigenfaces+KNN	39	66	96	96	33	30	60
Fisherfaces+KNN	94	100	97	100	100	97	64
3D Morphable Model	96	100	100	99	100	99	99
SOGF+MHD	97	100	100	100	100	100	99.5

100 updates were performed. The initial weights of all neurons were set to the greatest eigenvectors of the training data, and the neighborhood widths of the Gaussian function converged exponentially to 0.1 with the increase of time.

We utilized a fixed number of 4000 neurons, which arranged in different ways to give various map sizes (e.g., 100×40 , $40 \times 20 \times 5$, etc.). The results showed in Table 1 indicate that higher dimensional map seems to be more beneficial to the performance of the system. Intuitively, the choice of map dimension reflects the quality of the topographic order of neurons, that is, as the dimensions gets smaller or larger than the intrinsic dimensionality of the feature space, the topographic order of the map is destroyed and thus lose this important characteristic. 5-D SOM map seems to effectively represent the nonlinear manifold of the Gabor features caused by pose variations, it gives 99.5% accuracy for all tested poses given only one frontal image.

In the second experiment, a comparison between two classical linear feature extraction algorithms namely Eigenfaces (PCA), Fisherfaces (LDA) and one state of the art algorithm (3-D morphable model) is conducted. The experiment examine the performance of different feature extraction algorithms across pose changes. 1-KNN classifier algorithm based on Euclidean distance is employed for Eigenfaces and Fisherfaces algorithms while modular Hausdorff is used for Self-organized Gabor Features. Results shown in Table 2 reveal that global features extracted by PCA/LDA are highly sensitive to pose variations while local features extracted by SOGF is more robust. Furthermore, Fisherfaces algorithm outperform Eigenfaces by small margin due to its supervised learning capability. 3D morphable model algorithm considered as state of the art in face

Table 3. Performance of proposed algorithm using features from nose, mouth, and eyes regions

Component	c_{37}	c_{05}	c_{09}	c_{07}	c_{29}	c_{11}	Mean
Eyes	59	97	94	100	97	88	89
Nose	53	97	97	100	97	50	82
Mouth	88	100	97	97	97	78	93
All	94	100	97	100	100	91	97

recognition across pose due to its perfect capability to separate pose parameters and identity features. The weakness of the 3D model approaches is that they require 3D models and complicated fitting algorithms. Compared to the above, the proposed recognition scheme possesses the following advantages: (i) No manual selection for feature points are required; (ii) It is able to handle new pose conditions even if the algorithm given one Gallery image; (iii) No face alignment is required which considered as an essential preprocessing step in many algorithms.

The aim of the third experiment is to examine the contribution of each face component in the performance of the system under viewpoint changes. Recognition accuracy is calculated from the features captured by each component map separately. Roughly speaking eye features are the most discriminative features between subjects. However, the results shown in Table 3 reveal that features extracted from mouth region are the most invariant to pose variations followed by features from eye and nose regions. Furthermore, combining features from all component maps seems to be more effective than using only single component-map in the recognition.

4 Conclusion and Future Works

In this paper, we introduced a novel self-organized Gabor features based face recognition system across pose variations. Gabor wavelets first derive desirable facial features characterized by spatial frequency, spatial locality, and orientation selectivity. Self-organizing map method has been successfully captured the variations in the Gabor space caused by pose changes. Selecting the appropriate dimension of SOM has a great effect in the accuracy, which implies that each dimension in the map can be considered as a nonlinear principal component in the feature space. Local feature matching based on the Hausdorff distance metric proved to be robust to viewpoint changes compared with global feature extraction methods like Eigenfaces and Fisherfaces. Finally, the experimental results showed that features around mouth and eyes are the most discriminative and invariant features in comparison with features extracted from nose region. In the future, the effect of simultaneous variations in illumination and pose will be analyzed using proposed face recognition system.

References

1. Li, S., Jain, A.: Handbook of face recognition. Springer, New York (2005)
2. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. *Acm Computing Surveys* 35(4), 399–459 (2003)
3. Turk, M., Pentland, A.: Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 3(1), 71–86 (1991)
4. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 19(7), 711–720 (1997)
5. Wiskott, L., Fellous, J.M., Kruger, N., von der Malsburg, C.: Face recognition by elastic bunch graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 19(7), 775–779 (1997)
6. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23(6), 681–685 (2001)
7. Beymer, D.: Face recognition under varying pose. Technical report. MIT Press, Cambridge (1993)
8. Beymer, D., Poggio, T.: Face recognition from one example view. In: *Proceedings of Fifth International Conference on Computer Vision, 1995*, pp. 500–507 (1995)
9. Gross, R., Matthews, I., Baker, S.: Appearance-based face recognition and light-fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 26(4), 449–465 (2004)
10. Liu, C.J., Wechsler, H.: Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Trans. on Image Processing* 11(4), 467–476 (2002)
11. Blanz, V., Vetter, T.: Face recognition based on fitting a 3d morphable model. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 25(9), 1063–1074 (2003)
12. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(6), 643–660 (2001)
13. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust object recognition with cortex-like mechanisms. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 29(3), 411–426 (2007)
14. Kohonen, T.: *Self-Organizing Maps*. Springer, Heidelberg (1997)
15. Dubuisson, M., Jain, A.: A modified hausdorff distance for object matching. In: *Proc. 12th International Conference on Pattern Recognition*, pp. 566–568 (1994)
16. Sim, T., Baker, S., Bsat, M.: The cmu pose, illumination, and expression database. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 25(12), 1615–1618 (2003)

Image Hierarchical Segmentation Based on a GHSOM

Esteban José Palomo, Enrique Domínguez, Rafael Marcos Luque,
and José Muñoz

Department of Computer Science
E.T.S.I. Informatica, University of Malaga
Campus Teatinos s/n, 29071 – Malaga, Spain
{ejpalomo,enriqued,rmluque,munozp}@lcc.uma.es

Abstract. A novel approach for image segmentation is proposed in this paper. This approach is based on the growing hierarchical self-organizing map (GHSOM), which consists of a hierarchical architecture composed of growing self-organizing maps (SOMs). The SOMs have shown to be successful for the analysis of high-dimensional input data as in data mining applications. The hierarchical architecture of the GHSOM is more flexible than a single SOM in the adaptation process to input data, mirroring inherent hierarchical relations among input data. Image hierarchical segmentation can be achieved by using this neural network model, where the hierarchical structure of segmented regions is captured. In order to evaluate the performance of this segmentation method, an application for hierarchical background modeling in video sequences is provided. Therefore, foreground detection is achieved. Experimental results show that the proposed approach is promising for applications where hierarchical segmentation is required.

Keywords: Segmentation, data clustering, hierarchical self-organization, background modeling.

1 Introduction

Data clustering is an unsupervised learning method to discover most similar groups from input data, where data belonging to one group are most similar than data belonging to different groups according to a similarity measure. These methods are especially useful when information about input data is unavailable and input data are usually represented as feature vectors in a high-dimensional space.

The self-organizing map (SOM) has been widely used for knowledge discovery, data mining, detection of inherent structures in high-dimensional data and mapping these data into a two-dimensional representation space [1]. This mapping retains the relationship among input data and preserves their topology. The main advantage of this method is visual understanding of data structure. However, SOMs have some difficulties related to their fixed network architecture in terms of number and arrangement of neurons, which has to be defined in advance, and their lack of representation of hierarchical relations among input

data. The growing hierarchical SOM (GHSOM) was proposed in [2] to solve both limitations. This neural network model has a hierarchical architecture divided into layers, where each layer is composed of different single SOMs with adaptive architecture that is determined during the unsupervised learning process according to input data.

In this paper, the GHSOM model is used for hierarchical image segmentation. Image segmentation methods divide an image into several regions, where the contents of each region represent meaningful objects. Then, the segmentation results can be used for subsequent stages such as object recognition. Mathematically, most of these methods operate on the principle of minimizing the within-region variance, or other measures of internal homogeneity [3]. Different approaches are commonly used for this principle, ranging from threshold techniques, and boundary techniques, to region-based techniques and hybridized approaches [4,5].

The usefulness of the hierarchical segmentation based on GHSOM is shown with an application for foreground detection in video sequences. Here, the GHSOM is trained and tested with a set of frames from the PETS 2001 sequence dataset, which has been used to evaluate the performance of tracking and surveillance.

The remainder of this paper is organized as follows. Section 2 provides a description of the GHSOM model for hierarchical segmentation. In Section 3, a background model is built by using the GHSOM model to detect the foreground. In Section 4, some experimental results after evaluate the foreground detector with PETS 2001 dataset are presented. Section 5 concludes this paper.

2 GHSOM Model

The GHSOM has a hierarchical architecture composed of layers, which consist of several growing SOMs [6]. Initially, the GHSOM consists of a single SOM of 2x2 neurons, but this architecture is automatically adapted depending on the input patterns during the training. The SOM can grow by adding neurons until reach a certain level of detail in the representation of the data mapped onto the SOM. After growing, each neuron of the map has to be verified to see whether they are expanded or not. If the neuron has a bad representation of the data, it is expanded in a new map in the next layer of the hierarchy in order to provide a more detailed representation. Once training has finished, the GHSOM mirrors the inherent structure of the input patterns, improving the representation achieved with a single SOM. Therefore, each neuron represents a data cluster, where data belonging to one cluster are more similar than data belonging to different clusters.

The adaptive growth process of a GHSOM, is controlled by two parameters τ_1 and τ_2 , which are used to control the growth of a map and to control the hierarchical growth of the GHSOM, respectively. But this adaptation depends mainly on the quantization error of the neuron (qe). The qe is a measure of the similarity of data mapped onto each neuron, where the higher is the qe , the higher is the heterogeneity of the data cluster. The quantization error of the unit i is defined as follows

$$qe_i = \sum_{x_j \in C_i} \|w_i - x_j\| \quad (1)$$

where C_i is the set of patterns mapped onto the neuron i , x_j is the j th input pattern from C_i , and w_i is the weight vector of the neuron i .

Initially, the quantization error at layer 0 must be computed as given in (2), where w_0 is the mean of the all input data I .

$$qe_0 = \sum_{x_j \in I} \|w_0 - x_j\| \quad (2)$$

The initial quantization error qe_0 , measures the dissimilarity of all input data and it is used for the hierarchical growth process of the GHSOM together with the τ_2 parameter, following the condition given in (3). That is, the quantization error of a neuron i (qe_i) must be smaller than a fraction (τ_2) of the initial quantization error (qe_0) to be a leaf neuron. Otherwise, the neuron is expanded in a new map in the next level of the hierarchy, so the smaller is the τ_2 parameter chosen the deeper will be the hierarchy.

$$qe_i < \tau_2 \cdot qe_0 \quad (3)$$

When a new map is created, a coherent initialization of the weight vectors of the neurons of the new map is used as proposed in (7). This initialization provides a global orientation of the individual maps in the various layers of the hierarchy. Thus, the weight vectors of neurons mirror the orientation of the weight vectors of the neighbor neurons of its parent. The initialization proposed computes the mean of the parent and its neighbors in their respective directions.

A new map created from an expanded neuron is trained as a single SOM. During the training, the set of input patterns are those that were mapped onto the upper expanded unit. In each iteration t , an input pattern is randomly selected from this data subset. The winning neuron of the map is the neuron with the smallest Euclidean distance to the input pattern, whose index r is defined in (4).

$$r(t) = \arg \min_i \{\|x(t) - w_i(t)\|\} \quad (4)$$

The winner's weight vector is updated guided by a learning rate α , decreasing in time (5). In addition to the winner, the neighbors of the winner are updated depending on a Gaussian neighborhood function h_i and its distance to the winner. This neighborhood function reduces its neighborhood kernel in each iteration.

$$w_i(t+1) = w_i(t) + \alpha(t)h_i(t)[x(t) - w_i(t)] \quad (5)$$

When the training of the map m is finished, the growing of the map has to be checked. For that, the quantization error of each neuron (qe_i) must be computed in order to compute the mean of the quantization error of the map (MQE_m). If the MQE_m of the map m is smaller than a certain fraction τ_1 of the quantization error of the corresponding parent neuron u in the upper layer, the map stops

growing. This stopping for the growth of a map is defined in (6). Otherwise, the map grows to achieve a better level of representation of the data mapped onto the map, so the smaller is the τ_1 parameter chosen the larger will be the map.

$$MQE_m < \tau_1 \cdot qe_u \quad (6)$$

The growing of a map is done by inserting a row or a column of neurons between two neurons, the neuron with the highest quantization error e and its most dissimilar neighbor d . The neuron d is computed according to the expression (7), where Λ_e is the set of neighbor neurons of e .

$$d = \arg \max_i (\|w_e - w_i\|), \quad w_i \in \Lambda_e \quad (7)$$

3 Hierarchical Background Segmentation

GHSOMs can be used for image hierarchical segmentation by training this model with input data from images. These input data are represented by vectors of three features that represent each color component of a pixel, depending of the color space used (RGB, Lab, HSV, etc). In fact, just using the color information of each pixel, the pixel components will be mapped into a neuron so that each neuron will represent a set of pixels that are similar among them and the image will be segmented by colors, that is, each neuron represent a segmented region and each region can be hierarchical. In fact, if some neuron represents a set of heterogeneous pixels, it will be expanded at a subsequent layer of the hierarchy, creating a new map that will be trained with the pixels mapped onto the parent neuron. Non-expanded neurons are called leaf neurons and represent a region at the most level of granularity. Thus, an image hierarchical segmentation just using color information is provided. An example of hierarchical segmentation of an image and its generated GHSOM architecture is given in Fig. 11.

Image hierarchical segmentation using GHSOMs can have multiple applications. In order to evaluate this novel method, an application for hierarchical background modeling and foreground detection in video sequences has been implemented. Foreground detection can be achieved by analyzing the hierarchical information from the background model.

The hierarchical background model is based on hierarchical segmentation with GHSOM. A set of N frames from a video sequence is chosen, which are considered as background images of the scene. Then, the GHSOM is trained with the N background frames so that we obtain a hierarchical segmentation of the background. Thus, the generated GHSOM structure after training represents a hierarchical model of the background, where mappings between pixels of each frame and neurons are stored.

Let $k = (i, j)$ be the position of a pixel in a frame, N_k (8) be the set of mapped neurons of the pixel x_k and N_k^1 (9) be the set of layer 1 mapped neurons of the pixel x_k in the training. A pixel x_k belongs to foreground if its 1 layer neuron mapped n is different to all 1 layer neurons mapped in the training ($n \notin N_k^1$).

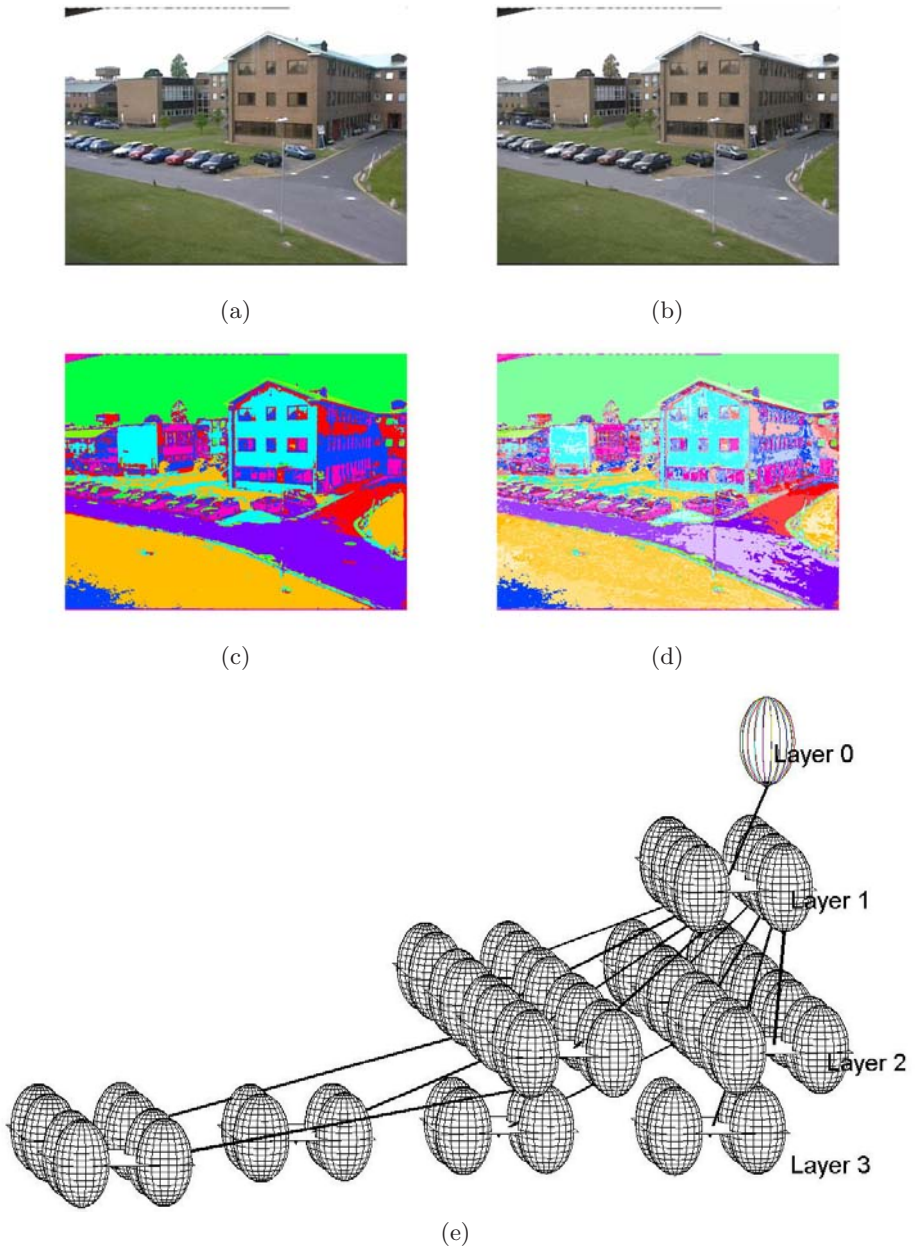


Fig. 1. An example of image hierarchical segmentation: (a) Original image. (b) Image where each pixel is represented by its associated leaf neuron weight vector. (c) Image segmentation at level 1. (d) Image where leaf neurons are represented by different colors, having similar colors leaf neurons with the same father. (e) Generated GHSOM architecture after training using the original image (a), with 3 layers, 8 neurons at layer 1, 47 leaf neurons and altogether 60 neurons.

$$N_k = \{i/x_k \text{ was mapped to } n_i\} \quad (8)$$

$$N_k^1 = \{i/x_k \text{ was mapped to } n_i \text{ at layer 1}\} \quad (9)$$

To increase the robustness of the detection criterion, the leaf neurons where the pixels are mapped can be analyzed, benefiting from hierarchy. Therefore, for pixels that satisfy the previous criterion, if the Euclidean distance between the pixel x_k and the weight vector of the mapped leaf neuron w_i is smaller than the Euclidean distance between x_k and the weight vectors of the mapped leaf neurons in the training w_j , the pixel belongs to foreground. This detection criterion is defined as follows

$$\|x_k - w_i\| < \|x_k - w_j\|, \quad \forall j \in N_k \quad (10)$$

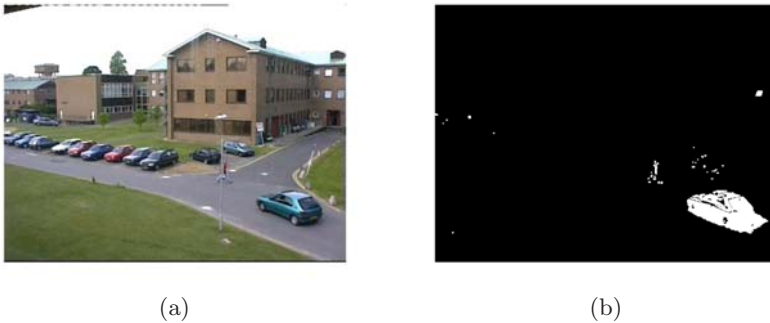


Fig. 2. (a) Original test frame. (b) Image with foreground objects detected.





4 Experimental Results

In order to evaluate the foreground detection system based on our hierarchical segmentation, PETS 2001 data set has been used¹. PETS 2001 consists of five separate sets of training and test sequences, where each set consists of one training sequence and one test sequence. Each frame of the sequence has a resolution of 576x768 pixels. In the segmentation phase, we chose 20 frames from the first dataset that we consider as background. Fig. 1(a) is one of them. 0.6 and 0.1 as values for τ_1 and τ_2 parameters, respectively. This way, we achieved a hierarchical architecture of 3 layers that was shown in Fig. 1(e).

Test frames were selected from the dataset to detect the foreground. These test frames include objects that did not appear in the training frames and are considered as foreground objects. Post-processing techniques have been applied to remove noise and improve the foreground detection. A test frame and the binary image with detected foreground objects are shown in Fig. 2.

¹ [Online] Available: <ftp://ftp.pets.reading.ac.uk/pub/PETS2001/>

Table 1. Foreground detection results with different trainings

				
False Positives	0.06%	0.3%	0.07%	0.03%
τ_1	0.6	0.4	0.6	0.6
τ_2	0.1	0.1	0.1	0.1
α	0.3	0.3	0.1	0.3
epochs	2	2	2	4
layers	3	4	3	3
1 layer neurons	8	12	9	8
overall neurons	60	72	61	52

Several experiments have been performed to compare the effects on foreground detection when GHSOM training parameters are modified and, therefore, hierarchical segmentation is different. These differences on foreground detection can be noted in Table 1. False positives are the percentage of background pixels that are detected as foreground. Note that the parameter τ_1 is the most decisive since control the size of the maps and, therefore, the depth/shalowness of the resulting hierarchical GHSOM. Thus, changing 0.6 to 0.4 as value for τ_1 parameter, the number of layers and neurons increase and the detection is worse. This is due to the number of neurons at the first layer, where the higher is the number of neurons the higher is the probability to make a mistake mapping a background pixel into a neuron. On the other hand, the lower the number of layer 1 neurons, the lower the foreground objects detection. Also, the number of epochs helps GHSOM learn better the input patterns and, therefore model the background of the video sequence.

5 Conclusions

In this paper, a novel approach for image hierarchical segmentation is proposed. This approach is based on the growing hierarchical self-organizing map (GHSOM), which is composed of several independent growing self-organizing maps (SOMs). The hierarchical architecture provides a more flexible adaptation process to input data and mirrors hierarchical relations among input data.

Hierarchical segmentation has multiple applications. An application to generate a hierarchical background model is proposed in order to detect the foreground and take advantage of this method. For our experiment, PETS 2001 data set has been used, where training frames considered as the background from a video sequence has been selected for training the GHSOM. Once training is finished, the generated architecture provides a hierarchical segmentation that represents the background of the video sequence. This way, when a new frame is presented to the trained GHSOM, comparison between the resulting mappings from pixels

into neurons with the mapping done during the training, provides information about whether classify a pixel as foreground or background. Since we have only used color components to segment the image, this hierarchical segmentation can be improved by adding more features to input pixels. Also, other applications can exploit hierarchical segmentation such as image compression, tracking and so on.

Acknowledgements. This work is partially supported by the Spanish Ministry of Science and Innovation under contract TIN-07362.

References

1. Kohonen, T.: Self-organized formation of topologically correct feature maps. *Biological cybernetics* 43, 59–69 (1982)
2. Rauber, A., Merkl, D., Dittenbach, M.: The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data. *IEEE Transactions on Neural Networks* 13, 1331–1341 (2002)
3. Beaulieu, J., Goldberg, M.: Hierarchy in picture segmentation: a stepwise optimization approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11, 150–163 (1989)
4. Fan, J., Yau, D., Elmagarmid, A., Aref, W.: Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Transactions on Image Processing* 10, 1454–1466 (2001)
5. Ohkura, K., Nishizawa, H., Obi, T., Hasegawa, A., Yamaguchi, M., Ohyama, N.: Un-supervised image segmentation using hierarchical clustering. *Optical Review* (2000)
6. Alahakoon, D., Halgamuge, S., Srinivasan, B.: Dynamic self-organizing maps with controlled growth for knowledge discovery. *IEEE Transactions on Neural Networks* 11, 601–614 (2000)
7. Dittenbach, M., Rauber, A., Merkl, D.: Recent advances with the growing hierarchical self-organizing map. In: 3rd Workshop on Self-Organising Maps (WSOM), pp. 140–145 (2001)

An Efficient Coding Model for Image Representation

Zhiqing Li^{1,2,3}, Zhiping Shi¹, Zhixin Li^{1,2}, and Zhongzhi Shi¹

¹ The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

² Graduate University of Chinese Academy of Sciences, Beijing 100190, China

³ College of Information Engineering, Xiangtan University, Xiangtan 411105, China
{lizq, shizp, lizx, shizz}@ics.ict.ac.cn

Abstract. The role of early sensory neurons is to remove statistical redundancy in the sensory input. In this paper, we propose a novel efficient coding model combining sparse coding strategy and selective attention strategy for image representation. The model is divided into two modules. In the first module, we employ the sparse coding strategy for natural image feature extraction. Furthermore, inspired by the selective attention strategy in biological visual system, we propose a self-adaptive algorithm to further reduce the activated variables in the second module. Compared with standard sparse coding (SC), the experimental results show that the efficient coding model evidently decreases the number of coefficients which may be activated and preserves the main structural information at the same time. Moreover, our model employs fewer responses to preserve similar perceptual image quality than other models.

Keywords: Image Representation, Sparse Coding, Structural Similarity, Selective Attention, Biological Visual System.

1 Introduction

Efficient coding hypothesis [1] provides a quantitative relationship between environmental statistics and neural processing. Barlow hypothesized that the role of early sensory neurons is to remove statistical redundancy in the sensory input. Furthermore, Olshausen and Field put forward a model, called sparse coding, which made the variables (or neurons stimulated by the same stimulus in the neurobiology.) be activated (i.e. significantly non-zero) only rarely [2, 3]. Vinje's result validated the sparse properties of neural responses under natural stimuli conditions [4]. Since then sparse coding theory was broadly investigated [5-8].

However, the number of variable which has a large value produced by sparse coding model and is possible to activated, is relatively large compared with the computation capacity of neurons, though the kurtosis of every response coefficient is also high. Thus, how to further reduce the activated variables in the same time to retain the important information as much as possible is very valuable in practice.

Another important problem is that objective methods for assessing perceptual image quality were the mean squared error (MSE), computed by averaging the squared

intensity differences of reconstructed and actual image pixels, along with the related quantity of peak signal-to-noise ratio (PSNR). This simplest and most widely used full-reference quality metric is appealing because it is simple to calculate, has clear physical meaning, and is mathematically convenient in the context of optimization. However, it is not very well matched to perceived visual quality [9].

In this paper, we propose a novel efficient coding model combining sparse coding strategy and selective attention strategy for image representation. First, we introduce structural similarity for quality assessment based on the assumption that human visual perception is highly adapted for extracting structural information from a scene. Then, employing the quality assessment method that takes advantage of known characteristics of the human visual system (HVS), we propose a self-adaptive algorithm to further reduce the activated variables.

The rest of this paper is structured as follows. In Section 2 we present the SC model and structural similarity. Section 3 describes a novel efficient coding model combining sparse coding strategy and selective attention strategy based on response saliency. Experiment results are reported and analyzed in Section 4. Finally, we conclude the paper in Section 5.

2 Sparse Coding Model and Structural Similarity

A perceptual system is exposed to a series of small image patches, drawn from one or more large images, just like the classic receptive field (CRF) of neurons. Imagine that each image patch represented by the vector I (numbered row-wise) has been formed by the linear combination of N basis functions. The basis functions form the columns of a fixed matrix, A . The weight of this linear combination is given by a vector, S . Each component of this vector has its own associated basis function, and represents a response value of a neuron in vision system. The linear synthesis model is therefore given by:

$$I(x, y) = AS = \sum_i a_i \Phi_i(x, y) \quad (1)$$

In a cortical interpretation, the S model the responses of (signed) simple cells, and the column of matrix A closely related to their CRF's.

2.1 Sparse Coding Model

Olshausen and Field applied two criteria to seek the optimal basis vector and the coefficients [3]. One of the criteria is how well the code describes the input. It was measured by the squared error between the input and its reconstruction by the network:

$$\text{Error}(A, S) = \sum_{x,y} \left[I(x, y) - \sum_i a_i \Phi_i(x, y) \right]^2 \quad (2)$$

As an additional criteria for sparse coding, Olshausen and Field proposed the 'sparseness' cost for seeking sparse codes.

$$\text{Sparseness}(A, S) = \sum_i S\left(\frac{a_i}{\sigma_i}\right) \tag{3}$$

where $S(x)$ is a nonlinear function such as $|x|$, $\exp(-x^2)$, and $\log(1+x^2)$. The cost sparseness favors the codes which consist of minimal number of non-zero coefficients. Thus, the search for a sparse code can be formulated as an optimization problem by constructing the following cost function to be minimized:

$$E(A, S) = \sum_{x,y} \left[I(x, y) - \sum_i a_i \Phi_i(x, y) \right]^2 + \lambda \sum_i S\left(\frac{a_i}{\sigma_i}\right) \tag{4}$$

Learning is accomplished by minimizing (4). The process for minimizing $E(A, S)$ can be divided into two nested stages. In the inner stage, $E(A, S)$ is minimized with respect to the a_i for a batch of pattern, holding the A fixed. In the outer stage (i.e, on a long timescale, over many image presentations), $E(A, S)$ is minimized with respect to the A .

I and Y denotes respectively actual and reconstructed images, Φ_i and a_i denotes respectively the i th column of A and the i th row of S , $\Phi_{i,j}$ denotes the element of A , λ is the weight of sparseness. Thus, the inner stage minimization over the a_i can be performed by conjugate gradient method, so the a_i is determined by the differential equation:

$$\bar{\nabla}_{a_i} E(A, S) = -2 \sum_{k=1}^N (I_k - Y_k) \phi_{k,i} + \frac{\lambda}{\sigma_i} S'\left(\frac{a_i}{\sigma_i}\right) \tag{5}$$

The outer stage minimization over the A may be finished by simple gradient descent method. The learning rule as follows:

$$\bar{\nabla}_{\phi_{i,j}} E(A, S) = -2(I_i - Y_i) a_j \tag{6}$$

$$\Delta \phi_{i,j} = -\eta \bar{\nabla}_{\phi_{i,j}} E(A, S) \tag{7}$$

where η is the learning rate.

2.2 Structural Similarity

Natural image signals are highly structured: their pixels exhibit strong dependencies, especially when they are spatially proximate, and these dependencies carry important information about the structure of the objects in the visual scene. Moreover, the HVS is highly adapted to extract structural information from the visual scene. Wang and Bovik et al. [10] developed a Structural Similarity Index and demonstrate it provides a good approximation to perceptual image quality through a set of intuitive examples.

The structural similarity between signals x and y is given by

$$SSIM(x, y) = \frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{8}$$

where $\mu_x = \bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$, $\mu_y = \bar{y} = \frac{1}{N} \sum_{i=1}^N y_i$, $\sigma_x = \left(\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right)^{1/2}$,
 $\sigma_y = \left(\frac{1}{N-1} \sum_{i=1}^N (y_i - \mu_y)^2 \right)^{1/2}$, $\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y)$, $0 < C_1, C_2 \ll 1$.

3 A Novel Efficient Coding Model

Early vision creates representations at successive stages along the visual pathway, from retina to lateral geniculate nucleus (LGN) to V1. Li and Shi et al. [11] put forward an attention-guided sparse coding model (AGSC), which adapts to the limited computation capability of neural system and improves the efficiency for the traditional sparse coding model. However, AGSC has drawbacks because MSE is not very well matched to perceived visual quality.

In this paper, in order to further reduce the activated variables and explain two key information bottlenecks along the visual pathway, we propose a novel efficient coding model divided into two modules. Functional diagram of the efficient coding model is shown in Fig. 1. At the beginning, retina performs a transformation of the natural image into a ‘retinal image’. The retinal image used as an input to the sparse coding module of the simple cell. Then, the selective attention module performs the selective attention based on response saliency.

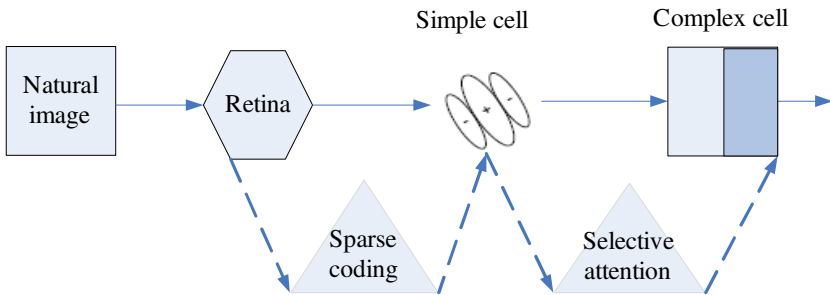


Fig. 1. Functional diagram of efficient coding model

3.1 Selective Attention Based on Response Saliency

Vision attention mechanism is an active strategy in information processing procedure of brain, which has many interesting characteristics, such as selectivity, competition. The simple cell’s response value and discrepancy distance based on their selective properties such as location, orientation and space frequency, formed the response saliency of simple cell [12]. The simple cells’ responses compete for being further processed in complex cell based on response saliency value.

Response saliency is to represent the conspicuity of every neuron in the same perception level for a stimulus and to guide the selection of attended neuron. The neuron response that has great response saliency value will be chosen to further process. On the contrary, the neuron that has small value will be omitted.

Intuitively, the response value itself provides very useful information: the response value is bigger, the information represented by the neuron is more important; otherwise, the information is less important. Obviously, the response value gives a foundation for the attention mechanism. Supposed here that A_i represents simple cell i , and R_i represents the simple cell's response. So R_i is greater, the response saliency value of A_i is also greater.

After we get the simple cell's response saliency values we can select certain simple cells as the complex cell's inputs according to the response saliency. Firstly, the simple cells responding to the stimulus are sorted by descend according to the response saliency value, and then a self-adaptive algorithm to further reduce and determine the number of activated variables.

Each image needs different number of variable to be activated according with itself. Inspired by the research of selective attention in psychology, we propose a self-adaptive algorithm combining sparse coding and selective attention. The self-adaptive algorithm consists of the following steps.

Algorithm 1. Combining sparse coding and selective attention

Input: a given threshold of structural similarity (inf_SSIM)

1. Extract image feature (i.e. responses of neurons) using SC
2. Responses are sorted by descend according to the response saliency value
3. Initialize number of responses to preserve ($N=8$)
4. Calculate the structural similarity between actual image and reconstructed image when

$$N \text{ responses were preserved } \text{SSim}(R', A) = \frac{1}{n} \sum_{k=1}^n \text{SSIM}(I_k(x, y), \sum_i R'_{i,k} A_i(x, y))$$

5. While ($\text{SSim}(R', A) < \text{inf_SSIM}$) do { $N=N+8$, goto step 4}
6. $N=N-7$
7. Calculate the structural similarity between actual image and reconstructed image when

$$N \text{ responses were preserved } \text{SSim}(R', A) = \frac{1}{n} \sum_{k=1}^n \text{SSIM}(I_k(x, y), \sum_i R'_{i,k} A_i(x, y))$$

8. While ($\text{SSim}(R', A) < \text{inf_SSIM}$) do { $N=N+1$, goto step 7}

Output: the number of responses to preserve (N)

4 Experiment Results

We conduct our experiments on a nature image data set which is available on the internet <http://www.cis.hut.fi/projects/ica/data/images/>. We sampled randomly sub-windows of 12×12 pixels 250000 times from original images, and converted every patch into one column. Thus, the input data set with the size of 144×250000 is acquired, and each image patch is represented by a 144 dimensional vector.

4.1 Image Feature Extraction

Using the updating rules of A and S as in (5) and (7) respectively in turn, we minimized the objective function given in (4). A stable solution was arrived at after 5000 updates (250000 image presentations). The learned basis functions (Fig. 2) simply reflects the fact that natural images contain localized, oriented structures with limited phase alignment across spatial frequency.

With learned basis functions, image feature extracting is accomplished by minimizing $E(A, S)$ with respect to the a_i for a batch of pattern, holding the A fixed. We sampled randomly sub-windows of 12×12 pixels 10000 times from original images, and converted every patch into one column. Using the above algorithm and 144 learned basis functions, image features of the input data set with the size of 144×10000 are extracted.

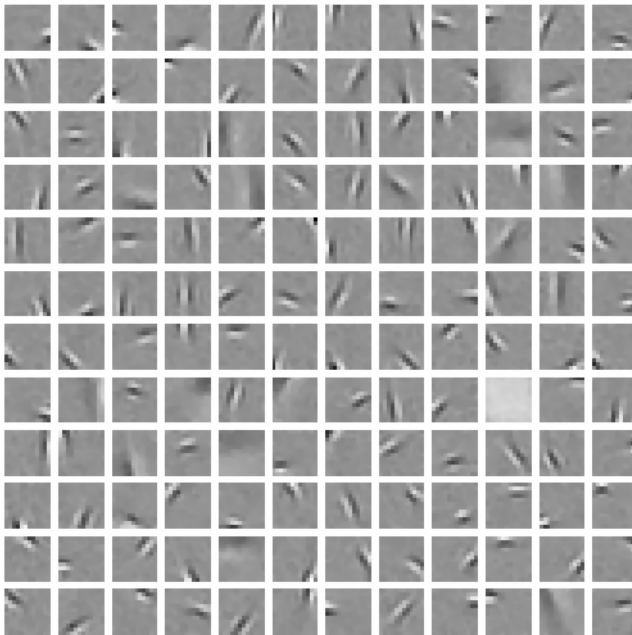


Fig. 2. Learned basis functions of sparse coding model

4.2 Image Reconstruction Using SC model and Selective Attention

In order to validate performance of our efficient coding model, we select Lenna (Fig. 3a) as test image. Lenna with 256×256 pixels was used widely in the image processing field.

The test image is randomly sampled 5000 times with 12×12 pixels to get the data set, and converted every patch into one column. Note that to find the accurate position of any image patch, we must remember the positions of each image patch appeared. Using the above 144 learned basis functions (Fig. 2), image features of the input data

set with the size of 144×5000 are extracted. The response coefficients produced by sparse coding model are mostly distributed around zero for each image patch of Leana.

In order to further reduce the number of activated variables in the same time to retain the important information as much as possible, algorithm 1 was employed to combine sparse coding and selective attention. The number of responses to preserve was determined by the algorithm, and then the reconstructed image was shown in Fig. 3b.

Because of sampling randomly, the same pixel might be founded in different image patches. Therefore, for the same pixel, we averaged the sum of the values of all reconstructed pixels, and used the averaged pixel value as the approximation of the pixel.



Fig. 3. (a) Original image. (b) Reconstructed image using our model. (c) Reconstructed image using AGSC model.

For comparison, we also used the AGSC model [13] to extract image features from the same data set, and the reconstructed image was shown in Fig. 3c. Percentage of responses to preserve for image representation using different models was shown in Table 1.

It is clear to see that reconstructed images (as shown in Fig. 3) are satisfying. It is difficult to tell reconstructed images from the original image only with naked eyes. However, comparison with the SC model, our method not only prominently reduces the number of activated coefficients, but also retains the main essential vision information for image representation. Furthermore, our model omits more coding coefficients to preserve similar perceptual image quality than AGSC model.

Table 1. Comparison of responses to preserve for image representation using different models

	AGSC-P	AGSC-T	Our model
Percentage of responses to preserve	45%	43%	27%

5 Conclusion

In this paper, in order to explain two key information bottlenecks along the visual pathway, we put forward an efficient coding model combining sparse coding strategy and selective attention strategy. Firstly, sparse coding strategy was employed for

simple cells to extract natural image features. Then, inspired by the research of selective attention in psychology, we propose a self-adaptive algorithm to further reduce the number of activated variables. By comparison with the SC model, the experimental result shows that our method not only prominently reduces the number of activated coefficients, but also retains the main essential vision information for image representation. Moreover, our model employs fewer responses to preserve similar perceptual image quality than AGSC model.

Acknowledgments

This work is supported by the National Science Foundation of China (No. 60933004, 60903141), the National Basic Research Priorities Programme (No. 2007CB311004), 863 National High-Tech Program (No.2007AA01Z132), and National Science and Technology Support Plan (No.2006BAC08B06).

References

1. Barlow, H.B.: Possible principles underlying the transformation of sensory messages. In: Rosenblith, W.A. (ed.) *Sensory Communication*, pp. 217–234. MIT Press, Cambridge (1961)
2. Field, D.J.: What is the goal of sensory coding. *Neural Computation* 6, 559–601 (1994)
3. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381, 607–609 (1996)
4. Vinje, W.E., Gallant, J.L.: Sparse coding and decorrelation in primary visual cortex during natural vision. *Science* 287, 1273–1276 (2000)
5. Grimes, D.B., Rao, R.P.N.: Bilinear sparse coding for invariant vision. *Neural Computation* 17, 47–73 (2005)
6. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research* 5, 1457–1469 (2004)
7. Hyvarinen, A., Hoyer, P.O.: A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images. *Vision Research* 41, 2413–2423 (2001)
8. Malo, J., Epifanio, I., Navarro, R., Simoncelli, E.P.: Non-linear image representation for efficient perceptual coding. *IEEE Transactions on Image Processing* 15, 68–80 (2006)
9. Wang, Z., Bovik, A.C., Lu, L.: Why is image quality assessment so difficult. In: *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 3313–3316. IEEE Press, New York (2002)
10. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 600–612 (2004)
11. Li, Q., Shi, J., Shi, Z.: A model of attention-guided visual sparse coding. In: *4th IEEE Int. Conf. on Cognitive Informatics*, pp. 120–125. IEEE Press, New York (2005)
12. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 1254–1259 (1998)
13. Li, Q., Shi, J., Shi, Z.: A selective sparse coding model with embedded attention mechanism. *Int'l Journal of Cognitive Informatics and Natural Intelligence* 4, 61–74 (2007)

SSTEM Cell Image Segmentation Based on Top-Down Selective Attention Model

Sangbok Choi¹, Sang Kyoo Paik², Yong Chul Bae², and Minho Lee^{1,*}

¹ School of Electrical Engineering and Computer Science, Kyungpook National University
1370 Sankyuk-Dong, Puk-Gu, Taegu 702-701, Korea

² Department of Oral Anatomy and Neurobiology, BK21, School of Dentistry,
Kyunpook National University, Daegu 700-412, Korea
choisb@ee.knu.ac.kr, {skpaik, ycbae, mholee}@knu.ac.kr

Abstract. We propose an automatic method for segmenting neurons in the TEM cell images based on a top-down attention model, which is efficient to solve the discontinuity problems in TEM cell image caused by loss of section or branching of cell. At first, the proposed model enhances cell boundaries using a partial differential equation based on hessian matrix, which can improve the contrast and continuity of cell membranes in the TEM images. Then, a top-down attention model trains the shape characteristics of the desired target neurons through the reinforcement and inhibition learning process. The top-down attention model localizes a candidate neuronal region in subsequent TEM image, which was implemented by a growing fuzzy topology adaptation resonance theory network (GFTART) model. It is efficient to resolve the discontinuity problem of TEM cell image. The localized candidate target neurons are finally indicated whether they are correct ones by an active appearance model (AAM). Experimental results show that the proposed method is efficient to segment the TEM images.

Keywords: serial-sectioning TEM (SSTEM), Top-down attention, cell image segmentation.

1 Introduction

In order to more precisely understand the brain's information processing mechanism, it might be really helpful to understand the structure of its neuronal circuit such as circuit interconnection topologies, the cell and synapse molecular that determine circuit signaling dynamics [1]. Electron microscopy (EM), which can provide resolutions on the order of 1 nanometer, remains the primary tool for resolving neurons, their sub-cellular 3D structures, and their synaptic connections.

Several of the high-resolution volume imaging methods generate large data set and spend long time to obtain the scanning images. For instance, the serial block-face SEM (SBFSEM), which is useful for the analysis of larger organism such as the blow-fly *Calliphora vicina* with a total brain volume of 1 mm³, takes about 21 years only for total scanning and 54 terabytes memory capacity [2]. However, it is possible to

* Corresponding author.

analyze only interesting regions of the brain which leads to a significant reduction of the time to acquisition and memory capacity. Therefore, a 3D reconstruction of neural circuit is performed only for each local neural circuit by recording one neuron at a time during processing by each local neural circuit. The morphology of an individual neuron might be useful for understanding in single cells [3].

There are still large data set that becomes enormous challenges of reliably abstracting biologically meaningful information about circuit connectivity and molecular architecture. Even though we can utilize the latest hardware and software for handling and tracing images, definitive results so far have been achieved only manual tracing, which is performed by human hand and eye [1]. Therefore, we need to develop a model for automated segmentation and discrimination of a neurobiologically meaningful object from the obtained scanning images. Some automated segmentation algorithms for the serial EM have been developed [1]. One approach for EM segmentation uses “Machine learning” algorithms, in which a program automatically optimizes its own operation based on “training sets” consisting of pairs of raw EM images and corresponding manual segmentation results [4]. In this approach, however, some difficulties are in producing sufficient and accurate training data sets for obtaining a robust and reliable performance by a machine learning algorithm. The other approach called “Contour-propagation” algorithm uses a semi-automated segmentation method, which is conducted by user interaction based on pixel-intensities of the current image in conjunction with the segmentation results of the previous image [2]. That approach assumes that objects are continuous across adjacent images which can be an appropriate method for processing continuous sequence of cell intensities and/or cell positions. Those two approaches have been applied to SBFSEM image data.

In our experiments, we try to analyze the very large serial-sectioning TEM (SSTEM) image data that has discontinuity properties, geometrical distortion and uneven section thickness. Therefore, both “Machine learning” and “Contour-propagation” algorithms are not enough for analyzing the SSTEM. To clear up those problems, we propose a new method that can segment a target cell region from candidate target areas in discontinuous sequence of the TEM cell image. Our proposed top-down attention model plays important role to find candidate area for segmentation. It will be a fully automatic method for segmenting neurons in the SSTEM cell images. Using top-down attention model, we take advantages both accuracy and processing time by localized candidate area.

This paper is organized as follows: in section 2, we describe our approaches to TEM image segmentation. In section 3, experimental results of the proposed model will be shown. Finally, we give brief summary and conclusion in section 4.

2 Proposed Model

In this paper, we propose an automatic method for segmenting neurons in the SSTEM cell images based on a top-down attention model. Figure 1 shows the overview of the proposed model. In preprocessing, the proposed model enhances cell boundaries using a partial differential equation based on a Hessian matrix, which can improve the contrast and continuity of cell membranes in the SSTEM images [5]. After preprocessing of the SSTEM cell images, a top-down attention model trains the shape characteristics of the desired target neurons through reinforcement and inhibition processes. The

top-down attention model was implemented by a growing fuzzy topology adaptation resonance theory network (GFTART) model [6]. The GFTART model is successively used to localize a candidate neuron region in subsequent SSTEM cell images, which not only enhance the segmentation accuracy but also reduce the computation load. Then, an active appearance model (AAM) is used to check whether the localized area by the top-down attention model includes a target neuron.

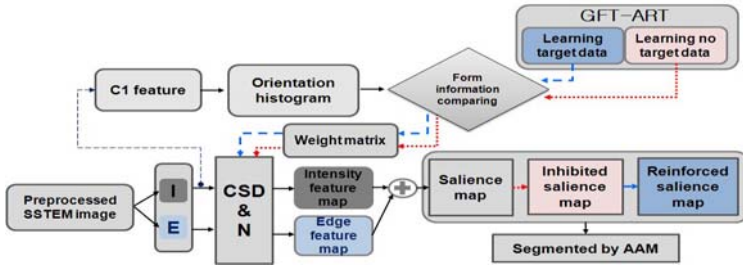


Fig. 1. Overview of proposed model

2.1 Preprocessing Step

It is very complicate to localize a neuronal area in the EM images. In order to obtain better localization, the histogram equalization is applied to the raw input intensity image of the EM, which improves the contrast of the neuronal membranes against the background. Moreover, in order to enhance the neuronal boundary among various stained textures, a partial differential equation based on a Hessian matrix [5] is applied to the histogram equalized EM images. Finally, we can preserve strong edge area in the EM images while smoothing weak edges substantially.

2.2 Top-Down Attention Model

When humans pay attention to a target object, the prefrontal cortex gives a competitive bias signal, related with the target object, to the infero-temporal (IT) and V4 area. Then, the IT and V4 area generates target object dependent information, and this is transmitted to the low-level processing part in order to make a competition between the target object dependant information and features in whole area in order to filter the areas that satisfy the target object dependent features.

The lower part in Fig. 2 generates a bottom-up saliency map (SM) based on primitive input features such as intensity and edge opponent [9-11]. In training mode, each salient object decided by the bottom-up SM is learned by a neural network called the GFTART [6]. The GFTART is implemented by integrating the conventional fuzzy ART, with the topology-preserving mechanism of the growing cell structure (GCS) unit [8]. In the GFTART, each node in the F2 layer of the conventional fuzzy ART network was replaced with GCS units [7]. Orientation histogram based C1 features in the hierarchical MAX model are used as form features [7]. For each object area, the log-polar transformed feature of Harris corner is used as a form feature. In top-down object biased attention, the GFTART activates one of memorized features according to task to find a specific object. The activated features related with a target object are

involved in competition with the features extracted from each local area in an input scene. By such a competition mechanism, as shown in Fig. 2, the proposed model can generate a top-down bias signal to a localized region for segmenting a target cell.

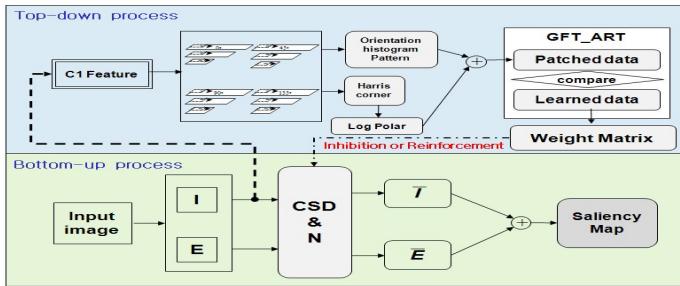


Fig. 2. Flow chart of top-attention model

Due to the top-down attention mechanism for a specific target cell, we can diminish the discontinuity problems caused by loss of section and branching of cells in TEM images since the top-down attention mechanism keeps more or less local key features of a target cell even when the shapes of the target cell varies through the successive sections.

2.2.1 Extraction of Form Features

Fig. 3 shows the feature extraction for an object area. Orientation features for 8 directions with 3 different scales are extracted from the intensity image of an input scene, which are called C1 features. Then we obtain 4 patches from 4 each different orientation feature, from which an orientation histogram with 4x4x8 dimension is generated. Each patch is divided by 4x4 sub-areas. An orientation histogram with 4x4 values for each patch is obtained using an average value of each sub-area.

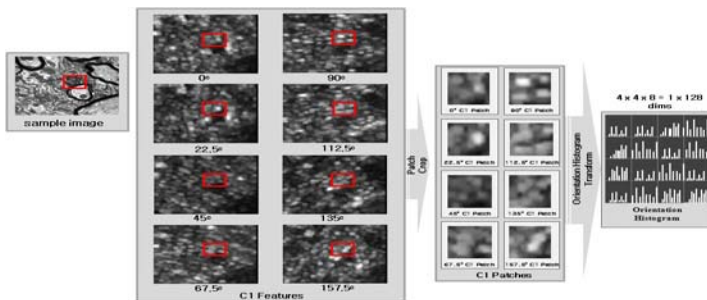


Fig. 3. Extraction of orientation histogram

Then a final orientation histogram with 4x4x8 dimensions is generated by concatenating 8 orientation histogram obtained from 8 different patches as shown in Fig. 3, then extract Harris corner from C1 features for considering of corner information as shown in Fig. 4.

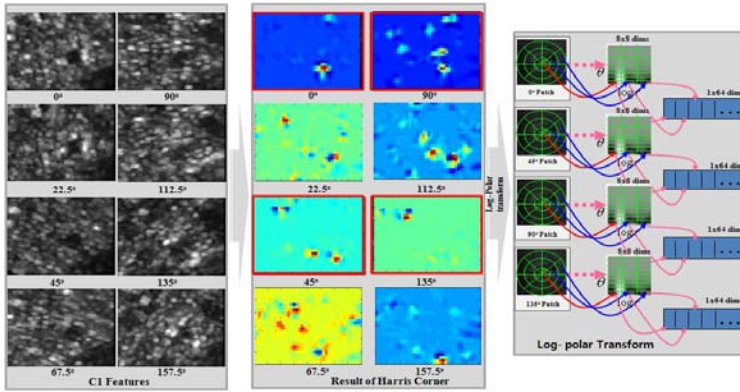


Fig. 4. Extraction of Harris corner

Corner information is made by 4 directions of C1 features against noise. 4 corner features are transformed by the log-polar process in order to make an input data with the same dimension regardless of different size of attention area. So we can construct a shape model with C1 feature and Harris corner, which mimics the information processing mechanism in V4 and IT.

Finally, we can make a vector by concatenating orientation histogram shown Fig. 3 and log-polar transformed corner features, which is used as input of the GFTART for from perception. And the size of input vector is 1×384 .

2.2.2 Growing Fuzzy Topology ART

The structure of the proposed GFTART is the similar with the convention fuzzy ART model. Instead, in the GFTART, each node in the F2 layer of the conventional fuzzy ART network was replaced with GCS units [7, 8]. The detailed GFTART algorithm is described in Fig. 5. The inputs of the GFTART consist of form features. Those features are normalized and then represented as a one dimensional array X that is composed of every pixel value a_i of the three feature maps and each complement a_i is calculated by $1 - a_i$, the values of which are used as an input pattern in the F1 layer of the GFTART model. Next, the GFTART finds the winning GCS unit from all GCS units in the F2 layer, by calculating the Euclidean distance between the bottom-up weight vector W_i , connected with every GCS unit in the F2 layer, and X is inputted. After selecting the winner GCS unit, GFTART checks the similarity of input pattern X and all weight vectors W_i of the winner GCS unit. This similarity is compared with the vigilance parameter ρ , which is the minimum these results similarity between the input pattern and the winner GCS. If the similarity is larger than the vigilance value, a new GCS unit is added to the F2 layer. In such situation, resonance has occurred, but if the similarity is less than the vigilance, the GCS algorithm is applied. The detailed GCS algorithm is described in [8].

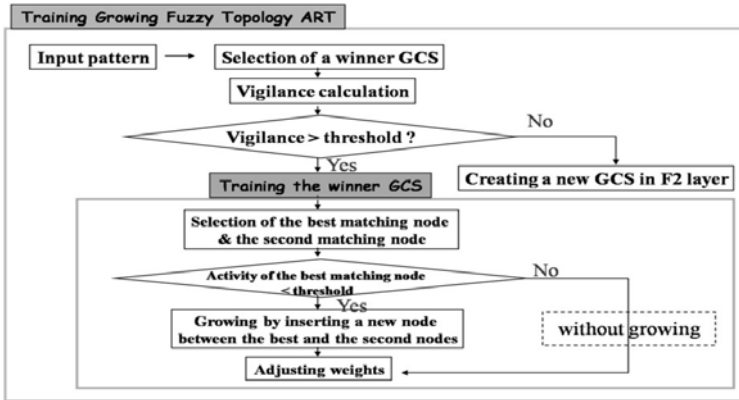


Fig. 5. Process flow of the growing fuzzy topology ART

The GFTART enhances the dilemma regarding the stability of fuzzy ART and the plasticity of GCS [8, 12]. The advantages of this integrated mechanism are that the stability in the convention fuzzy ART is enhanced by adding the topology preserving mechanism in incrementally changing dynamics by the GCS, while plasticity is maintained by the fuzzy ART architecture. Also, adding GCS to fuzzy ART is good not only for preserving the topology of the representation of an input distribution, but it also adaptively creates increments according to the characteristics of the input features.

2.2.3 Top-Down Biasing

In the proposed model, two GFTARTs are applied for training two different kinds of shapes of neuronal patterns. One GFTART memorizes target neuronal shapes (attractors) during training mode and reinforces a local area with memorized neuronal shapes of a target cell. The other GFTART memorizes non-target neuronal shapes (distractors) and inhibits a local area with memorized non-target neuronal shapes. Using two GFTARTs, the proposed model can efficiently localize the target areas among various types of similar patterns. After being successfully trained, the GFTARTs work for generating bias signals for reinforcing the target area and inhibiting the non-target area using similarity measure between the memorized patterns and the input patterns. In order to consider scale invariant characteristic of the top-down biased attention, 3 different scaled pyramids for shape features are considered. The bias signals generated for each local area constructs a weight matrix that uses as weighting factors for reinforcement and inhibition of a local area in the TEM images.

In order to enhance contrast between inner-side of boundary of the target area and outer-side of it, a Mexican hat function is applied to the local area originated at the center of the target area, f_m , as shown in Eq. (1), which generates the reinforcement weight matrix, W_r . In Eq. (2) f_s denotes neighbors of f_m . Moreover, in order to inhibit the non-target area, Gaussian function is applied to the local area originated at the center of the non-target area as shown in Eq. (2), which generates the inhibit weight matrix, W_n .

$$W_t = \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(fs-fm)^2}{2\sigma_1^2}\right) - \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{(fs-fm)^2}{2\sigma_2^2}\right) \quad (\sigma_1 < \sigma_2) \quad (1)$$

$$W_{nt} = -\frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(fs-fm)^2}{2\sigma_1^2}\right) \quad (2)$$

2.3 Segmentation by Active Appearance Model

The active appearance model (AAM) is applied to segment a neuronal pattern in a target area localized by the top-down attention model.

The technique of AAM has been introduced as a powerful technique for various automated medical image segmentation application [13, 14]. For training an AAM, all expert drawn contours obtained from manually annotated example images should have the same point distribution. A principal component analysis (PCA) is applied to the extracted contours in order to generate basis templates of neuronal contours. A PCA is also applied to the textures for generating basis templates of neuronal textures. Then, new neuronal structure basis templates are generated by each pair-wise combination of those two different kinds of basis. Moreover, the AAM model trains those obtained neuronal structure basis templates in order to get more generalized templates.

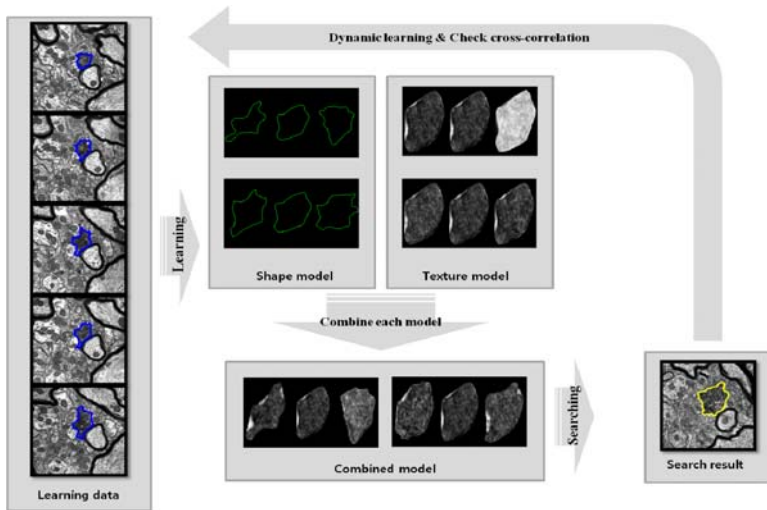


Fig. 6. AAM learning & searching flow

Fig. 6 shows the AAM process for localizing a specific neuron using trained neuronal structure templates. For accelerating fitting performance and speed we adapt dynamic learning because target object shape is varying.

As a final processing step, the cross-correlation between any two consecutive images is calculated. This allows the detection of failure of target segmentation and elimination of corrupted images when debris has gotten onto the section's face.

3 Experimental Result

Fig. 7 shows an experimental a top-down biased attention with preprocessed image. A target area becomes most salient area through inhibition and reinforcement weight generated by the GFTART model.

In order to show the effectiveness of the top-down attention model, we used an image stack obtained from primary afferents with presynaptic endings in the cat trigeminal interpolar nucleus. Serial ultrathin sections were collected on formvar-coated single slot nickel grids, counterstained with uranyl acetate and lead citrate, which are examined by an electron microscope.

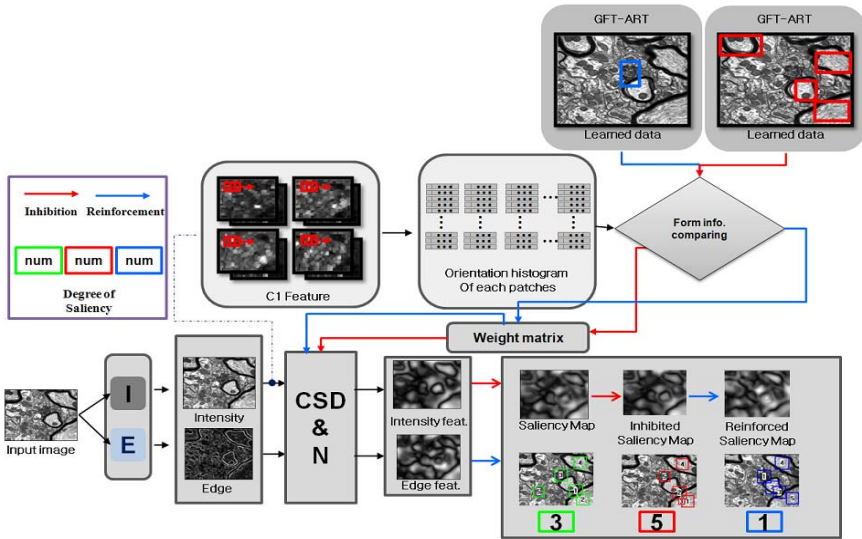


Fig. 7. The result of Top-down biasing in the proposed model

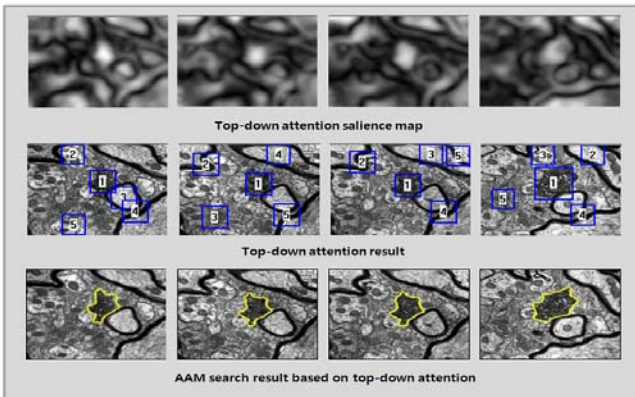


Fig. 8. The result of Top-down biasing in the proposed model

As shown in the middle row of Fig. 8, the proposed top-down attention model properly pop-outs a target area through inhibition and reinforcement of the GFTART model. And the AAM model also successfully segments the target area based on the area information localized by the top-down attention model as shown in the bottom row of Fig. 8. The top-down attention model correctly localizes by 90% and the AAM model successfully segments by 92%. As shown in Fig. 9, the AAM model based on top-down attention model generates better segmentation performance than the AAM based on the other models which was compared with the true restoration provided by a human expert.

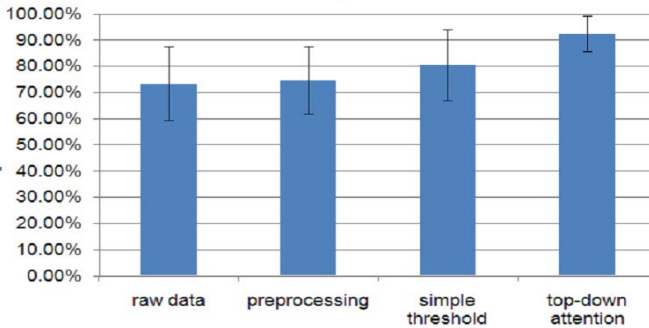


Fig. 9. Comparison of searching performance of AAM

4 Conclusion

We proposed a new neuron segmentation model based on the proposed top-down attention model and the AAM model, which shows plausible segmentation performance for the TEM images. The top-down attention mechanism successfully localizes a candidate area for a target cell, which helps to enhance the segmentation accuracy as well as computational efficiency. It can generate a reinforced bias signal when a local area has a resonance with the memorized shape pattern of a target cell. Also, it can keep local key features of a target cell even when the loss of section or branching of cells happens at the successive sections in the TEM images.

As further works, we need to test the proposed model using much more TEM images in order to verify generalization performance of the proposed model. Moreover, we need to verify the performance of the proposed model through comparing with that of another reconstruction algorithm of the TEM images.

Acknowledgments. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2009-0082262).

References

1. Smith, S.J.: Circuit reconstruction tools today. *Curr. Opin. Neurobiol.* 17, 601–608 (2007)
2. Macke, J.H., Maack, N., Gupta, R., Denk, W., Schoelkopf, B., Borst, A.: Contour-propagation algorithms for semi-automated reconstruction of neural processes. *J. Neurosci. Methods* 167, 349–357 (2008)

3. London, M., Hausser, M.: Dendritic computation. *Annu. Rev. Neurosci.* 28, 503–532 (2005)
4. Jain, V., Murray, J.F., Roth, F., Turaga, S., Zhigulin, V., Briggman, K.L., Helmstaedter, M.N., Denk, W., Seung, H.S.: Supervised learning of image restoration with convolutional networks. In: *ICCV* (2007)
5. Tasdizen, T., Whitaker, R., Marc, R., Jones, B.: Enhancement of cell boundaries in transmission electron microscopy images. In: *ICIP*, pp. 642–645 (2005)
6. Hwang, B., Lee, M., Ban, S.-W.: Top-Down Object Preferable Attention Using Growing Fuzzy Topology ART. In: *ICONIP*, pp. 284–285 (2008)
7. Riesenhuber, M., Poggio, T.: Hierarchical Models of object recognition in cortex. *Nat. Neurosci.* 2, 1019–1025 (1999)
8. Marsland, S., Shapiro, J., Nehmzow, U.: A self-organising network that grows when required. *Neural Netw. Special Issue* 15, 1041–1058 (2002)
9. Jeong, S., Ban, S.-W., Lee, M.: Stereo saliency map considering affective factors and selective motion analysis in a dynamic environment. *Neural Networks* 21(10), 1420–1430 (2008)
10. Goldstein, E.B.: *Sensation and perception*, 4th edn. An international Thomson publishing company, USA (1996)
11. Choi, S.B., Jung, B.S., Ban, S.-W., Niituma, H., Lee, M.: Biologically motivated vergence control system using human-like selective attention model. *Neurocomputing* 69, 537–558 (2006)
12. Carpenter, G.A., Grossberg, S., Makuzon, N., Reynolds, J.H., Rosen, D.B.: Fuzzy ART-MAP: A neural network architecture for incremental supervised learning of analog multi-dimensional maps. *IEEE Trans. Neural Netw.* 3(5), 698–713 (1992)
13. Oost, C., Koning, G., Sonka, M.: Automated contour detection in X-ray left ventricular angiograms using multiview active appearance models and dynamic programming. *ITMI* 25, 1158–1170 (2006)
14. Cootes, T., Taylor, C.: Anatomical statistical models and their role in feature extraction. *Br. J. Radiol.* 77, 133–139 (2004)

Data Partitioning Technique for Online and Incremental Visual SLAM

Nopparit Tongprasit, Aram Kawewong, and Osamu Hasegawa

Department of Computational Intelligence and System Science
Tokyo Institute of Technology, Japan

{tongprasit.n.aa,kawewong.a.aa,hasegawa.o.aa}@m.titech.ac.jp

Abstract. This paper describes a new data partitioning technique for used with a visual SLAM system. Combined with the existing SLAM system, the technique surveys areas to which the input image might belong to. It then retrieves matched images from such areas. The proposed technique can run in parallel with a normal SLAM system, such as FAB-MAP, in an unsupervised and incremental manner. We also introduce usage of Position-Invariant Robust Features (PIRFs) to make the system robust to dynamic changes in scenes such as moving objects. Combining our technique with normal SLAM can markedly increase the localization recall rate. Experiment results showed that the FAB-MAP result recall rate can increase to 30% at the same precision.

Keywords: Data partitioning, Visual simultaneous localization and mapping (visual SLAM), Invariant robust feature.

1 Introduction

Home-used robots are soon to be developed. Because of recent groundbreaking studies, home-used robots seem not to be mere daydreams anymore. Many researchers are striving to create humanoid robots that can assist humans in daily life in domestic tasks such as cooking, cleaning, and nursing. Nevertheless, without vision, even humans can barely finish such tasks by themselves. Robots cannot. Furthermore, how can a robot finish a task without knowing whether it is in a kitchen or living room? Consequently, Simultaneous Localization and Mapping (SLAM) system has become the focus of robot research and development (see [5] and [6] for reviews).

Existing SLAM systems share an important feature: a probability threshold for accepting or rejecting loop-closure detection. To achieve high precision, most systems must sacrifice lower recall for higher precision [1]. Somehow, in a highly dynamic environment, the obtained probability of some correct detected loop-closure might be too low to be accepted by the system. A high threshold guarantees 100% precision. However, it also rejects many correct loop-closure detections. This problem might be resolved if the system were able to change the threshold value for loop-closure acceptance or rejection adaptively, depending on the situation.

To achieve this capability, we propose a new technique that partitions the map into sub-maps, called *areas*. Using this technique, many correct loop-closure detections,

which were rejected by normal SLAM systems, could be accepted. Particularly, FAB-MAP performs accurate localization through image-to-image matching, whereas our system performs coarse localization through image-to-area classification. Consequently, some rejected loop-closures could be re-accepted if the current location could be determined to be somewhere in the map. Once it is confirmed that the currently detected loop-closure occurs at an area known to be visited previously, even a loop-closure with very low probability could be accepted by omitting the threshold.

This paper describes a specific examination of increasing the recall rate at "high" precision (i.e. 95–100% precision) because none of the practical SLAM systems can always perfectly offer precision of 100%. For an actual application, we might obtain only about 90–100% precision. Therefore, it would be useful to increase the rate of recall for every instance of decreased precision. As described herein, our proposed technique would be used with FAB-MAP. The entire system is tested on both City Centre and New College datasets described in an earlier report [1]. The system can match the location correctly even if the probability of the two images coming from the same location is 0.01 or less. Although the original FAB-MAP increase merely about 5–10% recall rate for 5–10% drops of precision, combining our technique with FAB-MAP increases the recall rate up to 30% for equally decreased precision.

2 Related Work

Simultaneous localization and mapping (SLAM) has been an important topic in robotics for nearly two decades [5], [6]. Failure in detecting loop closure based on metric data (metric SLAM) spurred researchers to present several appearance-based approaches to this task (visual SLAM). Despite their low cost, cameras can capture richer information than laser scanners or proximity sensors. Appearance-based methods can resolve perceptual aliasing problems, by which two places look similar.

Regarding visual SLAM, Bag-of-Words (BoW)-based approaches are considered state-of-the-art. Inspired by the Bag-of-Words image retrieval systems from the computer vision community [7], images are represented as a set of unordered elementary features (visual words) taken from a dictionary. The dictionary is built by clustering similar visual descriptors extracted from images into visual words. Using a given dictionary, images are classified by inferring their class based on the occurrence of words in an image. Images are represented as vectors of visual words' statistics with size equal to the number of words in the dictionary [9]. In fact, FAB-MAP [1]—which achieves the highest performance for this task over City Centre and New College datasets—presents a disadvantage in its offline process for dictionary generation. Angeli et al. [2] proposed the incremental dictionary for use in online applications. However, the accuracy of [2] is described as less than or equal to [1].

A remaining disadvantage of FAB-MAP is its low recall rate at 100% precision. We can improve the recall rate of FAB-MAP markedly at the same precision using a novel post-processing technique that runs in real-time with FAB-MAP. Our technique uses another visual feature that is especially robust against dynamic changes. Results obtained using our technique show marked improvement of the recall rate of localization at a high rate of precision, especially at around 90–95% precision.

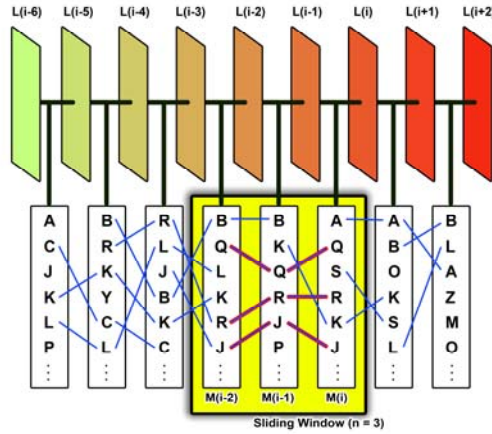


Fig. 1. Concept of Position-Invariant Robust Feature extraction. Alphabet characters represent each local feature. The same characters represent similar features. At time i , $P(i)$ contains features Q, R, and J.

3 Online Data Partition Technique

a Position-Invariant Robust Feature

Position-invariant Robust Feature, designated as PIRF, is developed upon the SIFT feature [8]. To be compatible with City Centre and New College datasets presented in an earlier report [1], the location L in this work includes 2 associated images: left-hand side and right-hand-side images. Figure 1 portrays a typical PIRF extraction. First, the system extracts SIFT features from the current location L_i and compares with SIFT features from L_{i-1} . Matched SIFT features are kept in the list of index $M(i)$. Given that n is the number of location per sliding window, the system finds similar SIFT features among $M(i-n+1)$, ..., $M(i-1)$, $M(i)$ in the same sliding window and keeps it in $P(i)$. Consequently, $P(i)$ represents a set of SIFTs which appear in the location from L_{i-n} to L_i . By repeating this process, the system can obtain "slow-moving" keypoints of L_i : the keypoints move slowly relative to the changes of the camera's positions. In many cases, PIRFs capture distant objects. It might be said that $P(i)$ belongs to objects that exist in all n locations L_{i-n} to L_i .

b Bag of Words Recognition

Bag of words [2] is another concept we have applied in our system. Raw SIFTs are too noisy to use directly to model the appearance of location. The costs for processing numerous SIFTs are too great. Therefore, Sivic and Zisserman [7] suggested clustering SIFTs into k clusters and their use as "visual words" to model the appearance. To apply this concept with PIRF, a single PIRF is used simply as a single word in the dictionary. Each area has its own dictionary. That is to say, if there are currently N areas in the map, there would be N associated dictionaries to represent the appearance

of each particular area. During localization, the system searches through all dictionaries and finds an area with the highest number of words matched the input image. The matching method is done as described in a previous report [8]. However, localization based on only the highest matches might cause errors. For example, the winner area has 20 matches and the runner-up area has 19 matches. This case is considered an unclear winning condition. To handle this appropriately, we perform the winning quality check. The *quality* is defined in eq. (1). In this work, the best threshold for *quality* is 1.2. Some correct match is judged as unclear winning; it decreases the recall rate of localization if the threshold is too high. Alternatively, if the threshold is too low, it decreases the precision rate. The system will reject and decide a location as unknown because the bag-of-words system has no confidence in its answer if the *quality* is lower than 1.2. Those unknown locations need additional information to localize from an Online Data Partition system.

$$quality = \frac{winner_score}{runner_up_score} \quad (1)$$

c Online Data Partition

This section presents a detailed description of our proposed technique. Two main tasks exist for SLAM: Localization and Mapping. A previously visited location must be localized to some place in the map. Otherwise, such a location is a new previously unseen location and must be mapped into the map as a new location. To do so, the system needs to localize the input image to some existing areas in the map first. The localization is confirmed as the correct localization if the localization results are apparently stable or in the same area. In contrast, if the results are confusing, it could be inferred that none of the mapped areas are a good match for the input image and that the image should therefore belong to the new area.

Particularly if location L_i belongs to some past area k , designated as A_k , the area of locations L_{i-1} and L_{i+1} should be A_k . However, in practice, if the areas of locations L_{i-1} and L_{i+1} differ from A_k , those situations can be classified into three cases: incorrect localization, entering another visited area, and entering new area.

In the first case, incorrect localization, if L_{i-1} and L_{i+1} are in the same area, L_i must be in the same area as well.

In the second case, entering another visited area, if the previous area of from L_{i-k} to L_{i-1} is A_m and area of L_i change to A_k and areas of L_{i+1} , L_{i+2} , L_{i+3} , ... remains A_k , system can conclude that, from location L_i , the system left area A_m and entered area A_k .

In the third case, that of entering a new area, because that area's dictionary has not been generated yet, no suitable dictionary exists for that location. This causes the system to localize areas of L_i , L_{i+1} , L_{i+2} , L_{i+3} , ... to different areas. Those locations cannot be localized into one single area. Instead, they scatter in a random manner. The system defines a new area from PIRF features of those locations.

Figure 2 presents the online data partition method algorithm. Given area k of location L_i represented as A_k , the system determines the most suitable area for L_i . Let *prev_area* be the area of L_{i-1} . Both L_{i-1} and L_i are in the same area. The system gives a reward to *confidence* in the prediction module if *prev_area* and A_k are the same with

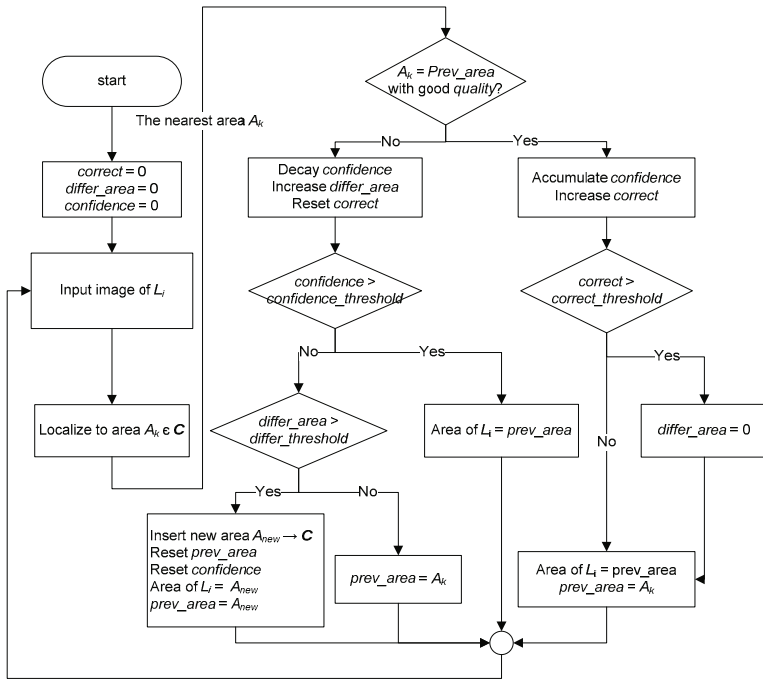


Fig. 2. Online data partition method. The variable *correct* helps the system decide when to reset *differ_area*.

quality beyond threshold. However, if *prev_area* differs from A_k , then the *confidence* gets a penalty and increases the number of different areas: *differ_area*. The *differ_area* represents the number of locations which do not belong to *prev_area*.

After assigning a penalty to *confidence*, the system checks both *confidence* and *differ_area*. The values of both variables can engender the following three cases.

Case I: Incorrect localization

The current location is judged as a misclassification if *confidence* is beyond the *confidence* threshold. The system sets the current area to *prev_area* and remembers the current location as the *start_point*. The *start_point* will be used if the location is not misclassified data, but is from some other location. The system can recover the result after retrieving additional information from the next location.

Case II: Entering another visited area

The system left *prev_area* and entered a visited area at *start_point* if both *confidence* and *differ_area* are lower than their thresholds. Behavior of the *confidence* value is dropping slightly before rising. The *differ_area* value will increase for some time but remain under the threshold because *prev_area* has changed to the other area. After localizing correctly for some time, the *differ_area* will be reset to 0.

Case III: Entering new area

The current location is in an unvisited area because areas from the locations sequence are unpredictable if *confidence* is lower than *confidence_threshold* and *differ_area* is greater than *differ_threshold*. We can assume that the current area has no dictionary. In this case, the system will gather PIRF features from those locations and generate a new dictionary. After generating the new area, the area of locations from L_{start_point} to current L_i is set to the created new area. Then *differ_area* is set to 0 and the *confidence* gets a reward ($i-start_point+1$) times.

The reward–penalty method described in this paper can be any mathematical function. In this work, we suggest the linear accumulation function as a reward function and exponential decay function as a penalty function because the system might lose sensitivity to the new area if we choose the same mathematical function.

4 Experiment and Result

As described herein, two major experiments were conducted. Experiment 1 was designed to examine the precision rate and recall after post-processing compared to the SLAM system. Experiment 2 showed the influence of threshold value on precision. The SLAM system used in this experiment is the FAB-MAP system [1], which is claimed to represent the state-of-the art.

Experiment 1: Precision and Recall Rate after Post-Processing

This experiment was designed to show the improvement of the recall rate after post-processing. Post-processing yields an unknown or new location of FAB-MAP’s result and finds the best answer in the area of that location. Post-processing tested both the City Centre dataset and New College dataset.

Figure 3 presents precision–recall curves of our post-processing comparing to the FAB-MAPs for City Centre and New College. According to Table 1, the recall rate at precision 0.95 can improve 31% in City Centre and 20% in New College.

Figures 4(a) and 4(b) respectively depict results at precision 0.95 of post-processing and FAB-MAP. The yellow dots show the path. Red dots show the loop-closure detection. Green lines connect loop-closure detected locations. Although at precision 0.95, our system located mistaken location to location near the correct location but not included in the Ground truth, while FAB-MAP pointed to a completely different location.

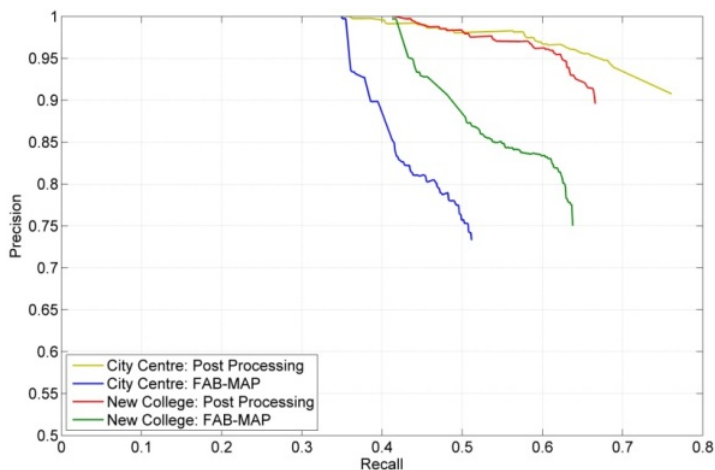
Experiment 2: Effect of Threshold on Precision

This experiment is designed to elucidate the influence of threshold on precision. The threshold in the post-processing system was set by discarding the probability of a same area, which is lower than specified during post-processing. Figure 5 portrays the relation between the threshold, precision, and recall rate.

From figure 5, it is apparent that decreasing the threshold in post-processing has less influence than in FAB-MAP. At this point, we can strongly assert that our system can answer correctly even the match between two locations that have low probability. In other words, our system depends only slightly on a threshold value.

Table 1. Comparison of results at different precisions on both after post-processing and FAB-MAP

Method	Precision	Recall: City Centre	Recall: New College
FAB-MAP	100%	34.94%	41.26%
Post-processing	100%	36.01%	42.44%
FAB-MAP	95%	36.01%	43.26%
Post-processing	95%	66.93%	62.37%

**Fig. 3.** Precision-recall curves on City Centre and New College dataset**Fig. 4.** Result for the City Centre dataset overlaid on an aerial photograph at precision 0.95. Figure 4(a) shows the result after post-processing. Figure 4(b) shows the result of FAB-MAP.

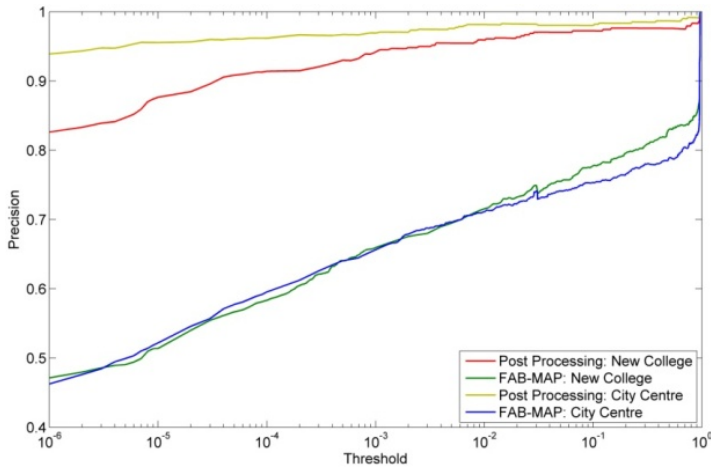


Fig. 5. Relation between threshold and precision

5 Discussion and Conclusions

This paper introduced post-processing for SLAM system. Our experiments demonstrated that our system can enhance the SLAM result. Although the system showed no significant improvement at precision 1, the system yields outstanding results with a high recall rate by sacrificing a small amount of precision.

However, our work is based on the SLAM system. This post-processing might not show marked improvement if the SLAM system fails to match the images of location containing highly dynamic change. Good normalization is necessary for this post-process as well.

References

1. Cummins, M., Newman, P.: FAB-MAP: Probabilistic Localization and Mapping in the space of Appearance. *Int'l. Jour. Robotics Research* 27(6), 647–665 (2008)
2. Angeli, A., Filliat, D., Doncieux, S., Meyer, J.A.: Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words. *IEEE Trans. Robotics* 24(5), 1027–1037 (2008)
3. Kawewong, A., Tangruamsub, S., Hasegawa, O.: Wide-Baseline Visible Features for Highly Dynamic Scene Recognition. In: *Proc. Int'l. Conf. Computer Analysis of Images and Patterns, CAIP* (2009)
4. Valgren, C., Lilienthal, A.: Incremental Spectral Clustering and Seasons: Appearance-Based Localization in Outdoor Environments. In: *Proc. IEEE Int'l. Conf. Robotics and Automation, ICRA* (2008)
5. Durrant-Whyte, H., Bailey, T.: Simultaneous Localization and Mapping: Part I. *IEEE Robotics & Automation Magazine* 13(2), 99–110 (2006)
6. Bailey, T., Durrant-Whyte, H.: Simultaneous Localization and Mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine* 13(3), 108–117 (2006)

7. Sivic, J., Zisserman, A.: Video Google: A Text Retrieval Approach to Object Matching in Videos. In: Proc. IEEE Int'l. Conf. Computer Vision, ICCV (2003)
8. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. Int'l. Jour. Computer Vision (IJCV) 60(2), 91–110 (2004)
9. Wang, J., Zha, H., Cipolla, R.: Coarse-to-fine Vision-Based Localization by Indexing Scale-Invariant Features. IEEE Trans. System, Man & Cybernetics 36(2), 413–422 (2006)

Improvement of Image Modeling with Affinity Propagation Algorithm for Semantic Image Annotation

Dong Yang and Ping Guo

Image Processing and Pattern Recognition Laboratory
Beijing Normal University
Beijing 100875, China
d.yang@ieee.org, pguo@ieee.org

Abstract. Semantic image annotation can be viewed as a classification problem, which maps image features to semantic labels, through the procedures of image modeling and image-semantic mapping. In order to improve the performance of image modeling, we propose a novel method which is based on affinity propagation (AP) algorithm. For a given image, low-level image features are extracted from image sub-blocks, and the image feature distribution can be modeled by a mixture of Gaussian components. An adaptive mixture component number selection algorithm which is related to the image semantic information is also developed. The AP algorithm is adopted to improve the efficiency and accuracy of the distribution estimation. For a given label, the overall distribution is modeled, and the mixture component number is selected according to the mixture exemplars extracted from all images and the average value of the preference parameter. The experiment results illustrate that the proposed algorithm has the higher efficiency and accuracy compared with C-means and expectation-maximization (EM) algorithm combination.

Keywords: image annotation, clustering, image modeling, affinity propagation algorithm.

1 Introduction

Automatic semantic image annotation is the process that the database of images are annotated with semantic labels by a computer system automatically. Semantic image annotation can be viewed as a mapping procedure from image features to semantic labels, by the steps of image modeling and image-semantic mapping. Image features include low-level visual features (color, shape, texture, topology), object-level features and 3-dimension scene features. While semantic labels include feature semantics, object semantics, scene semantics, behavior semantics and emotion semantics [1]. The low-level visual features have been successfully used in content based image retrieval (CBIR) [2]. However, high-level image features and semantic labels used in semantic based image retrieval (SBIR) [3] make the retrieval process more flexible.

To bridge the semantic gap between low-level image features and high-level semantic labels, we should focus on two key steps: image modeling and image-semantic mapping.

For image modeling, low-level image features are extracted from image sub-blocks, then the image feature distribution is represented by the Gaussian mixture model (GMM), for example, the model parameters are computed by C-means and EM algorithm combination [4] [5].

For image-semantic mapping, there are two categories of methods. If each semantic label is considered as a class, the mapping can be viewed as a semantic classification problem, such as earlier indoor-outdoor [10], blobworld [4] and supervised multiclass labeling (SML) [5] [11] problems. If each semantic word is viewed as a variable, the mapping is a image-semantic joint modeling problem, such as N-cut based method [3], latent dirichlet allocation (LDA) method [12] and cross-media relevance models (CMRM) [13]. Besides, relevance feedback methods integrate users' feedbacks to retrieve images [14].

When image-semantic mapping is viewed as the classification problem, semantic labels are considered as predefined classes, and the mapping is taken as supervised classification. Supervised OVA (one vs all) adopted two-class classifiers to learn from positive and negative images, while the positive images have the given semantic label and the negative images do not have [15]. Luo and Savakis [10] have approached the scene classification using a divide-and-conquer strategy, a good first step of which is to consider only two classes such as indoor and outdoor images, while the latter may be further subdivided into city and landscape images. SML [5] [11] adopted a multiclass Bayesian classifier to classify the images with multiple semantic labels, and assumed that the labels have independent distributions although each image has multiple labels. EM algorithm was adopted to iteratively estimate the distribution parameters. However, this is a computational expensive process. Affinity propagation (AP) clustering algorithm is to identify a relatively small number of features, called exemplars to represent the whole features [7] [8]. It seems to produce a better fitness function than mixture modeling with C-means methods [9].

In this work, we intend to apply AP algorithm to find out how to fast estimate the image density distribution model parameters, and how to efficiently produce image annotation results more precisely.

2 Methods

The framework of the proposed method is shown as Figure 1.

In Figure 1, rectangles represent objects, while rounded corner rectangles represent methods. To bridge the semantic gap between image features and semantic labels is the central target of semantic image annotation. And image modeling (modeling of one image), image-semantic mapping (modeling of images and supervised classification) are the three key steps.

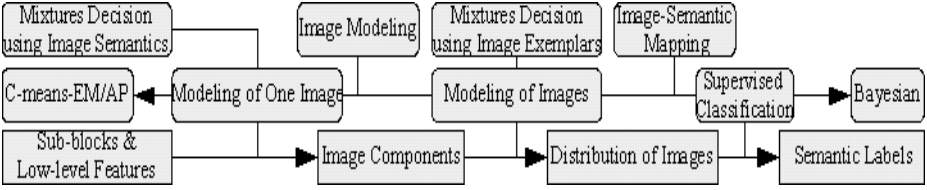


Fig. 1. Framework of the proposed method

2.1 Image Features

Considerable research efforts have been devoted to the low-level image features used in CBIR and SBIR. A localized color feature, which is the discrete cosine transform (DCT) coefficient vector of 8×8 image sub-blocks that overlap 6 pixels between adjacent blocks in YCbCr color space [5], is selected.

$$\mathbf{R}_{i,j}^c = \mathbf{I}^c(2i : 2i + 7, 2j : 2j + 7), \tag{1}$$

$$\mathbf{T}^c = \text{DCT}(\mathbf{R}^c) \quad c = y, cb, cr \quad i, j = 0, 1, 2, 3, \dots, \tag{2}$$

$$\mathbf{X} = [\mathbf{T}^y(\cdot)', \mathbf{T}^{cb}(\cdot)', \mathbf{T}^{cr}(\cdot)']', \tag{3}$$

$$f(\mathbf{I}) = \{\mathbf{X}_{0,0}, \mathbf{X}_{0,1}, \dots, \mathbf{X}_{1,0}, \mathbf{X}_{1,1}, \dots\}. \tag{4}$$

Where $\mathbf{R}_{i,j}^c$ is the (i, j) -th sub-block of image \mathbf{I} , \mathbf{T}^c is the DCT coefficient matrix of the sub-block \mathbf{R}^c , $\mathbf{X}_{i,j}(\cdot)$ is the feature vector that concatenates feature vectors from three color channels, and $f(\mathbf{I})$ is the set of image feature vectors.

2.2 Modeling of One Image

AP algorithm. AP algorithm can be applied to identify a relatively small number of exemplars to represent the whole feature vectors. Each feature vector is viewed as a node in a network, and real-valued messages are recursively transmitted along edges of the network until a good set of exemplars and corresponding clusters emerges. It can be briefly described as following [7]:

$$s(i, k) = - \|\mathbf{X}_d - \mathbf{X}_k\|^2, \tag{5}$$

$$r(i, k) \leftarrow s(i, k) - \max_{k' \neq k} \{a(i, k') + s(i, k')\}, \tag{6}$$

$$a(i, k) \leftarrow \min\{0, r(k, k) + \sum_{i' \neq i, i' \neq k} \max\{0, r(i', k)\}\}. \tag{7}$$

Where the similarity $s(i, k)$ indicates how well the feature vector with index j is the exemplar of feature i . The responsibility $r(i, k)$ reflects the accumulated

evidence for how appropriate feature k is the exemplar of feature i , considering other potential exemplars of feature i . Availability $a(i, k)$ reflects the accumulated evidence for how appropriate it would be for feature i to choose feature k as its exemplar, considering the support from other feature vectors that feature k should be an exemplar. When the preference $s(k, k)$ grows big, each node tends to select itself as the exemplar, then the number of clusters will increase [7].

Clustering features and the mixture model. Considering the dimension and amount of image features, the Gaussian mixture representation is compact and robust. Instead of C-means and EM algorithm combination, we propose an AP-based algorithm for image modeling:

- 1) AP algorithm is adopted to cluster the feature vectors into several groups with corresponding exemplars;
- 2) For each group, these similar feature vectors are used to estimate the Gaussian distribution. The weight of each group is estimated according to the number of feature vectors in the group;
- 3) Each image is represented by the mixture model of these Gaussian distributions and weights.

$$\{\mathbf{e}_i\} = \text{AP}(f(\mathbf{I}), p), \quad i = 1..cn \quad (8)$$

$$\mu_i = \mathbf{e}_i, \quad \Sigma_i = \text{cov}(\mathbf{A}_i), \quad \omega_i = \text{num}(\mathbf{A}_i), \quad (9)$$

$$\mathbf{A}_i = \{\mathbf{x} | \text{exemplar}(\mathbf{x}) = \mathbf{e}_i\}, \quad (10)$$

$$P_{\mathbf{X}|\mathbf{I}} = \sum_{i=1..cn} \omega_i G(\mu_i, \Sigma_i) \quad (11)$$

Where the parameter $f(\mathbf{I})$ is the set of image feature vectors. The preference parameter p can be estimated by the adaptive mixture component number selection algorithm described in the next section. And cn is the real number of the exemplars computed by AP algorithm. \mathbf{A}_i means the set of feature vectors whose representation exemplar is \mathbf{e}_i , and $G(\mu_i, \Sigma_i)$ means the Gaussian distribution with mean vector μ_i and covariance matrix Σ_i . $P_{\mathbf{X}|\mathbf{I}}$ is the mixture model of image \mathbf{I} .

An adaptive mixture component number selection algorithm. There have been several mixture component number selection principles, such as fixed number [5] and the minimum description length principle [4], or more general criterion [6]. We found that the mixture model of clustering features can be referred from the semantic information of the image. That is to say, instead of fixed or homogeneous component number, we develop an adaptive mixture component number selection method incorporating with the semantic labels of the image and corresponding label attributes.

$$cn(\mathbf{I}) = \sum_{s_i \in labels(\mathbf{I})} cn(s_i). \tag{12}$$

Where $cn(\mathbf{I})$ is the mixture component number of image \mathbf{I} , $labels(\mathbf{I})$ are the semantic labels of image \mathbf{I} . $cn(s)$ is the empirical approximate mixture number of the semantic label s , which can be estimated in advance. The mixture numbers of some semantic labels are shown as in Table 1.

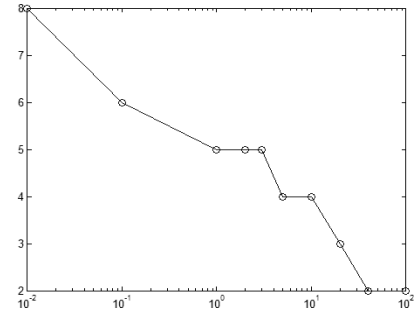
Table 1. Cluster number of several semantic classes

sky	plant	aeroplane	land	animal
1	1	1,2	2,3	2,3

In AP algorithm, the mixture component number is a variable that relies on the preference parameter. We find that there is a similar mapping relationship between preference and mixture number. As it is shown in Figure 2, from 20p to 100p all can lead to a two-class clustering result. This illustrates that there is a wide range of preference value that can produce a steady clustering result.



(a) original image



(b) cluster number vs. logarithm of the preference

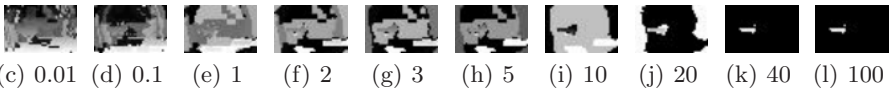


Fig. 2. Preference values influencing the clustering result. The original image and result images with increasing preference values(0.01p, 0.1p, ...), where p is the median similarity.

An empirical map between mixture number and preference value can be built up in advance. Taking the aeroplane picture 2(a) as an example, this picture has two labels: sky and aeroplane. By looking up the empirical approximate mixture number table, this picture might contain two or three clusters totally. Then by looking up the preference and mixture number map, the preference value might be 20 to 40. We can select a average value 30 as the preference for AP algorithm.

$$p = \text{map}(cn(\mathbf{I})) \quad (13)$$

The preference and mixture number map can be built up in the training process.

2.3 Modeling of Images

The goal of modeling images is to find the prior distribution and the class-conditional distribution in feature space, which can be computed from images with the given label.

Hierarchical distribution estimation. For a given label, images with this label contain two categories of features: features that belong to this label and features that do not belong to this class. There is an assumption that the former features tend to cluster together, while the latter features tend to spread over the entire feature space [5]. We believe that this assumption is reasonable when the number of samples of each label is large and balanced enough.

However, it is too expensive to estimate the distribution from all images at the same time. A hierarchical mechanism is adopted based on a mixture hierarchy where children densities consist of different combinations of subsets of the parents' components. A general description of bottom-up propagating parameters in two consecutive levels is given using EM algorithm [16].

The semantic label distribution can be estimated using the mixture model of images.

$$\mathbf{A}_s = \{P_{\mathbf{X}|\mathbf{I}}|s \in \text{labels}(\mathbf{I})\}, \quad (14)$$

$$P_{\mathbf{X}|s} = H(\mathbf{A}_s) = \sum_{i=1, \dots, cn(s)} \omega_i G(\mu_i, \Sigma_i). \quad (15)$$

Where $P_{\mathbf{X}|\mathbf{I}}$ is the mixture model of image \mathbf{I} computed in the section 2.2 and $P_{\mathbf{X}|s}$ is the distribution of label s . \mathbf{A}_s represents the distribution of images with label s , and the function $H(\cdot)$ is the hierarchical distribution estimation algorithm. Fixed cluster number is required when applying $H(\cdot)$ to build mixture model, therefore we need to find out the largest number of clusters from all images, and supply null components to those mixture models that have less number of clusters.

A class-level mixture component number selection algorithm. The mixture component number of class-conditional distribution can be inferred from the exemplars of this class, because the number of the exemplars is relatively smaller than that of all feature vectors. For the hierarchical distribution estimation, this selected number adapts to the real distribution than that of fixed number.

$$cn(s) = cn(\mathbf{A}), \quad \mathbf{A} = \{\text{exemplars}(\mathbf{I})|s \in \text{labels}(\mathbf{I})\}. \quad (16)$$

Where $cn(s)$ is the mixture number of label s , and $s \in \text{labels}(\mathbf{I})$ means all images with label s .

The algorithm is as follows:

1) For each image in the class, the exemplars and preference parameters are recorded after AP clustering (section 2.2).

2) The average value of the preference parameters is used in clustering the exemplars, in order to produce a proper mixture component number.

The image-level and class-level mixture component number selection algorithms are different:

1) For a given image, the former algorithm adaptively computes an component number instead of fixed number. And a mixture number and preference map is built up previously, because the AP algorithm requires preference parameter instead of component number.

2) For a given label, the latter algorithm adopts AP algorithm to compute the component number instead of fixed number, which is required as a parameter in hierarchical distribution estimation algorithm.

2.4 Supervised Classification

Under the framework of Bayesian classification, both the image annotation and retrieval can be implemented with a minimum probability of error principle. For a given class, the probability that a test image belongs to this class is the product of the class-conditional probabilities of the image components.

$$\lg(P_{\mathbf{I}|s}(\mathbf{I}|s_i)) = \sum_{\mathbf{X} \in \mathbf{I}} \lg(P_{\mathbf{X}|s}(\mathbf{X}|s_i)) \quad (17)$$

By introducing a set of class-conditional distributions, the semantic annotation results for this image can be obtained with the labels whose posterior probabilities ($P_{s|\mathbf{I}}(s_i | \mathbf{I})$) are the first several large values.

3 Experiments

In this section, we validate the efficiency and accuracy of the image modeling with AP algorithm through annotation results. The images are selected from database [17] and [18]. We selected a subset of outdoor images which contain five classes: aeroplane, sky, land, plant and animal, altogether 378 images are selected. Typically each image contains three or more classes. In order to speed up the processing, all images are resized to the small blocks with the size of from 200×200 to 300×300 pixels.

Table 2 illustrates the efficiency of the proposed algorithm.

The experiment procedure is described as follows:

1. Half of images are for training set, and half of those for testing. Six different sets of training images are selected and the adjacent two sets have five-sixth overlap.
2. For each training set, three factors are computed: a) percentage of some attractive label annotated; b) percentage of all labels annotated; and c) percentage of any wrong label annotated (Figure 3).

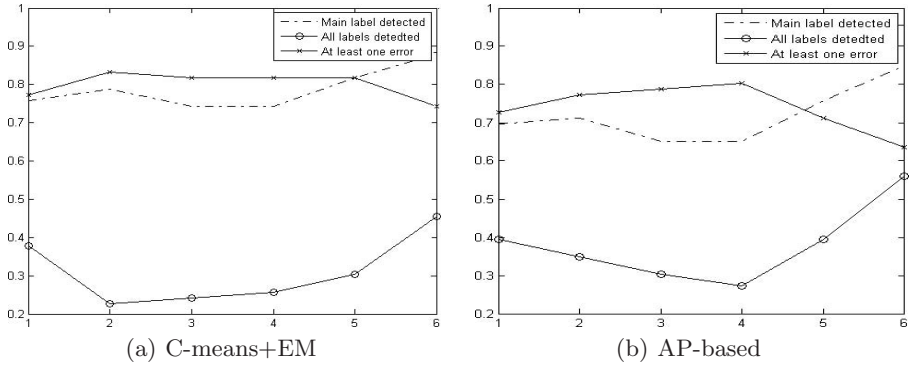


Fig. 3. Two algorithms are compared using the percentage of images which are labeled correctly

Table 2. Time consumption of modeling one image

methods	25 loops max	50 loops max	100 loops max
C-means	25.8	43.3	46.5
C-means+EM	63.6	133.7	167.3
AP-based	17.8	45.4	73.6

3. For each label, recall and precise factors are computed, averaging from all six training sets (Figure 4).
4. The average time consumption is computed.

For a given semantic descriptor, assuming that there are w_H human annotated images in the test set and the system automatic annotates number is w_{Auto} , of which w_C are correct, recall and precision are given as following:

$$recall = \frac{w_C}{w_H}, \tag{18}$$

$$precise = \frac{w_C}{w_{Auto}} \tag{19}$$

The labels that are manually annotated might relate with obscure features of the image. Comparing the C-means and EM algorithm combination with the AP-based algorithm, we find that there are about 70 % of test images in which most attractive label is annotated, and about 30 % of images in which all labels are annotated. However, AP-based algorithm improves the percentage that all labels are annotated, and reduces the percentage that wrong label is annotated(Figure 3).

Figure 4 illustrates that the accuracy is improved with the proposed algorithm for three classes, while that for other two classes is near same with C-means and EM algorithm combination. From Figure 3 we can easily know that the recall or precise values are different when the classification model is built with different training sets, which means that the distributions of the labels in the database are

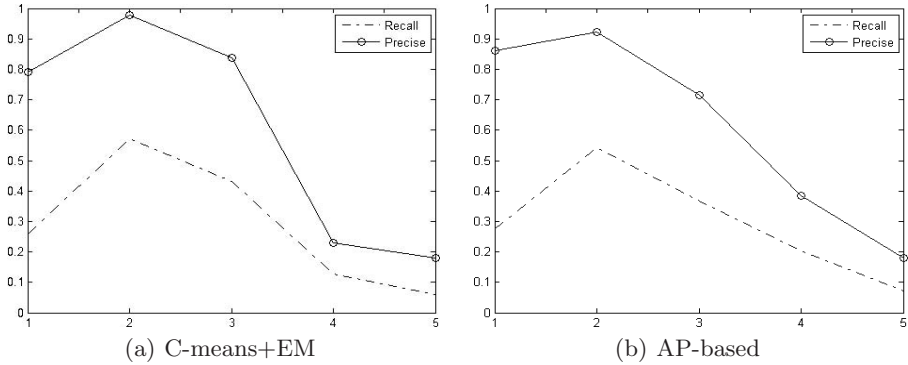


Fig. 4. The comparison of the average recall and precise values, computed with one particular label among five labels

uneven. The recall or precise values of the five classes in Figure 4 are different, probably because the classes have large difference amount of information in those images. That is to say, the image database requires to be well organized in order to improve the annotation performance.

4 Conclusions

In this paper, we have investigated the improvement problem of image modeling with AP algorithm for semantic image annotation. The efficiency and accuracy of distribution estimation is improved when AP algorithm is adopted. For a given image, a mixture component number selection method is developed on considering the semantic labels. For a given label, the mixture component number is selected according to the average parameter value and the mixture exemplars extracted from all training data set. The experiment results show that the effectiveness of the developed number selection methods. When the algorithm developed from this study is applied to the automatic image annotation problem, it certainly can accelerate and optimize the image retrieval process.

Acknowledgement. The research work described in this paper was fully supported by the grants from the National Natural Science Foundation of China (Project No. 60675011, 90820010). Prof. Ping Guo is the author to whom all correspondence should be addressed.

References

1. Eakins, J.P.: Automatic Image Content Retrieval - Are We Getting Anywhere? In: Proc. of the Third International Conf. on Electronic Library and Visual Information Research, pp. 123–135 (1996)
2. Datt, R., Li, J., Wang, J.Z.: Content-based Image Retrieval: Approaches and Trends of The New Age. In: ACM SIGMM International Workshop on Multimedia Information Retrieval, pp. 253–262 (2005)

3. Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D.M., Jordan, M.I.: Matching Words and Pictures. *J. of Machine Learning Research* 3, 1107–1135 (2003)
4. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Blobworld: Image Segmentation Using Expectation-maximization and Its Application to Image Querying. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24(8), 1026–1038 (2002)
5. Carneiro, G., Chan, A.B., Moreno, P.J., Vasconcelos, N.: Supervised Learning of Semantic Classes for Image Annotation and Retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 29, 394–410 (2007)
6. Guo, P., Chen, C.L.P., Lyu, M.R.: Cluster number selection for a small set of samples using the Bayesian ying-yang Model. *IEEE Trans. Neural Network* 13(3), 757–763 (2002)
7. Frey, B.J., Dueck, D.: Mixture Modeling by Affinity Propagation. In: *Advances in Neural Information processing Systems*, pp. 379–386 (2006)
8. Frey, B.J., Dueck, D.: Clustering by Passing Messages between Data Points. *Science* 315, 972–976 (2007)
9. Dueck, D., Frey, B.J.: Non-metric Affinity Propagation for Unsupervised Image Categorization. In: *IEEE International Conf. on Computer Vision*, pp. 1–8 (2007)
10. Luo, J., Savakis, A.: Indoor vs Outdoor Classification of Consumer Photographs Using Low-level and Semantic Features. In: *International Conf. on Image Processing*, pp. 745–748 (2001)
11. Vasconcelos, N.: Minimum Probability of Error Image Retrieval. *IEEE Trans. on Signal Processing* 52(8), 2322–2336 (2004)
12. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. *J. of Machine Learning Research* 3(5), 993–1022 (2003)
13. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic Image Annotation and Retrieval Using Cross-media Relevance Models. In: *Annual ACM Conf. on Research and Development in Information Retrieval*, pp. 119–126 (2003)
14. Zhou, X.S., Huang, T.S.: Relevance Feedback in Image Retrieval: A Comprehensive Review. *Multimedia Systems* 8(6), 536–544 (2003)
15. Vailaya, A., Figueiredo, M., Jain, A., Zhang, H.J.: Image classification for content-based indexing. *IEEE Trans. on Image processing* 10(1), 117–130 (2001)
16. Vasconcelos, N.: Image Indexing with Mixture Hierarchies. In: *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol. 10, p. I-3–I-10 (2001)
17. Visual Object Classes Challenge, <http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2008>
18. Caltech 256, http://www.vision.caltech.edu/Image_Datasets/

An Image Identifier Based on Hausdorff Shape Trace Transform

Rerkchai Fooprateepsiri¹, Werasak Kurutach¹, and Sutthipong Tamsumpaolerd²

¹ Department of Information Technology

² Department of Computer Engineering

Mahanakorn University of Technology

51 Cheum-Sampan Rd., Nongchok, Bangkok, 10530

{rerkchai, Werasak, Sutthipo}@mut.ac.th

Abstract. This paper presents a robust method for digital image identification under conditions of variant illumination, compression, flip, scaling, rotation and gray scale conversion. Techniques introduced in this work are composed of two parts. The first one is the signature of image is to be detected by the Trace Transform [6]. Then, in the second part, the notion of Hausdorff distance [8] and Modified Shape Context [10] are employed to measure and to determine the similarity between the models and tested images. Finally, our approach is evaluated with experiments on a set of over 60,000 unique images and one billion images pairs. The experimental result has show that the average of accuracy rate is higher than 83%.

Keywords: Image Retrieval, Image Identification, Hausdorff Distance, Trace Transform.

1 Introduction

Large numbers of image databases now exist that contain multiple modified versions of the same image. An extreme example of this is the large number of modified versions of images on the internet (web site). There is a need to develop tools that will enable the identification of all of the original and modified versions of the same images. Identification of image in image databases has a challenging problem despite of over three decades of research efforts. This is because the identifier must be robust to common image processing modifications such as rotation, scaling, grayscale conversion, compression, blur, and Gaussian noise. In the other requirements are that the descriptor should be compact, it should not be excessively expensive for extraction and it must allow very fast searching. Image identifiers are also known by the terms image hashes [1], image signatures [2] and image fingerprints [3].

However, there are several areas that are related to image identification. Although these areas are all related they are somewhat different in their requirements. The first, image similarity, involves looking for images that are perceptually similar in some sense. The solution to similarity matching can be more relaxed about the results returned in terms of the false acceptance. The work in area of image area of image identifiers can be broadly classified into three approaches by their support region, i) local feature point based [4], ii) region based [5] and iii) global [2].

- *Local feature point based methods have the undesirable characteristic that they have high complexity in terms of searching. This is a result of the need to compare all points from one image with all point in another image*
- *Region-based approaches overcome some of the complexity problems associated with feature-based approaches they suffer from a lack of invariance to geometrical transformations. Region-based approaches perform particularly poorly in the presence of significant rotation.*
- *Global support region methods have shown some promise in terms of search complexity and robustness. One such method exploits the invariant properties of the Fourier-Mellin transform. Whilst this method shows some interesting results it uses principal components analysis on a set of training images which leads to the signature being specific to a particular dataset.*

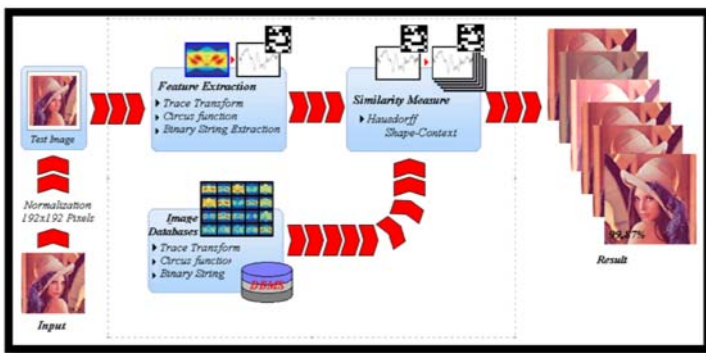


Fig. 1. An Image Identification System

A number of methods based on line projection in images have been proposed. In [1] lines are projected through a centre point in the image to form a 180 sample feature vector. The DCT components of the feature vector are taken and then quantized to form an identifier. Matching is carried out using a peak cross-correlation method. The concept of the radial projections is similar to a method based on the Radon transform [2]. The Radon transform of the image is taken and then a number of steps including a 2D FFT are performed to extract a 2D 20x20 binary identifier for an image.

Our approach is similar to [2], however there are several significant and beneficial differences. We use the more general Trace transform, rather than the Radon transform, allowing multiple component identifiers to be extracted. Also the intermediate steps are less computationally demanding, the 2D FFT is no longer necessary and a 1D FFT can be used. Lastly, the method presented here uses fewer bits for the image identifier which results in lower storage requirements and faster searching. The organization of this paper is as follows. Section 2 introduces a method for tracing line on an image and some trace functional we used in this paper. We introduce a shape matching measure in section 3. In section 4, we present our experimental results. Finally, we conclude in section 5.

2 Feature Extraction

2.1 Trace Transform

The Trace transform [6], a generalization of the Radon transform, is a new tool for image processing which can be used for recognizing objects under transformations, e.g. rotation, translation and scaling. To produce the Trace transform one computes a functional along tracing lines of an image. Each line is characterized by two parameters, namely its distance ρ from the centre of the axes and the orientation ϕ . The normal to the line has with respect to the reference direction. In addition, we define parameter t along the line with its origin at the foot of the normal. The definitions of these three parameters are shown in figure 2. The image is transformed to another image with the Trace transform which is a 2-D function depending on parameters (ϕ, ρ) . Different Trace transforms can be produced from an image using different trace functional. An example of the Trace transform is shown in figure 3. It is shown that the image space in the x and y directions is transformed to the Trace transform space in the ϕ and ρ directions.

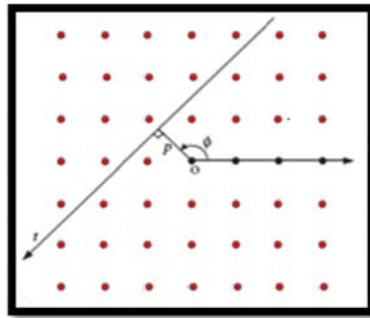


Fig. 2. Tracing line on an image with parameters ϕ, ρ and t

The key property of the Trace transform is that it can be used to construct features invariant to rotation, translation and scaling. We should point out that invariance to rotation and scaling is harder to achieve than invariance to translation. Let us assume that an object is subjected to linear distortions, i.e. rotation, translation and scaling. It is equivalent to saying that the image remains the same but viewed from a linearly distorted coordinate system. Consider scanning an image with lines in all directions. Let us denote the set of all these lines with Λ . The Trace transform is a function g defined on Λ with the help of T which is some functional of the image function when it is considered as a function of variable t . T is called the *trace functional*.

$$g(\phi, \rho) = T(F(\phi, \rho, t)), \tag{1}$$

where $F(\phi, \rho, t)$ stands for the values of the image function along the chosen line. Parameter t is eliminated after taking the trace functional. The result is therefore a 2-D function of parameters ϕ and ρ and can be interpreted as another *image* defined

on Λ . The resultant Trace transform depends on the functional we used. Let us denote $t_i \in t$ the sampling points along a tracing line defined by ϕ and ρ . Let us also denote by n the number of points along the tracing line. n may be varied depending on the length of the tracing line. The trace functionals used in our experiments are:

$$T(f(t)) = \int_0^\infty f(t) dt \tag{2}$$

$$T(f(t)) = [\int_0^\infty |f(t)|^\rho dt]^q \tag{3}$$

$$T(f(t)) = \text{median}_t\{f(t), |f(t)|\} \tag{4}$$

The denomination $\text{median}_x\{x, w\}$ means the weighted median of sequence x with weights in the sequence. For example, $\text{median}\{\{4,2,6,1\},\{2,1,3,1\}\}$ indicates the median of numbers 4,2,6 and 1 with corresponding weights 3, 1, 2 and 1. This means the standard median of the numbers 4, 4, 2, 6, 6, 6, 1, i.e. the median of the ranked sequences 1, 2, 4, 4, 6, 6, 6 is 4. (See [6]. for more details and the properties of the Trace transform).

A further functional can then be applied to the columns of the Trace transform to give a 1D function of the angle ϕ . This second functional is known as the diametrical functional and the resulting function is known as the circus function. Two different diametricals are applied to obtain the circus functions in Figure 4. The properties of the circus function can be controlled by appropriate choices of the two different functionals (trace and diametrical).

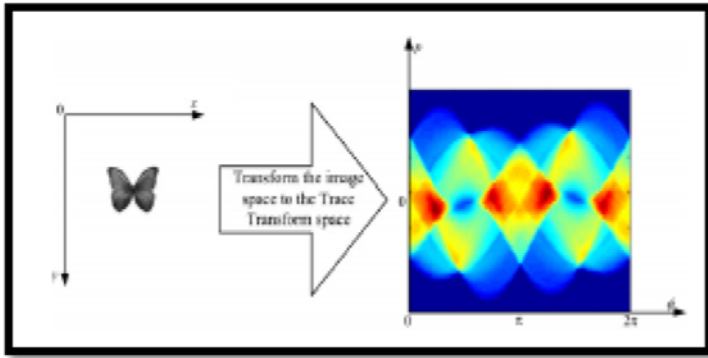


Fig. 3. An image and its Trace Transform

For rotation, scaling and translation it can be shown that [10] with a suitable choice of functionals the circus function $c(a)$ of image a is only ever a shifted or scaled version of the circus function $c(a')$ of the modified image a'

$$c(a') = Kc(a - \theta) \tag{5}$$

The property of (5) is exploited in [3] to obtain an object signature and it is also used here to obtain a visual identifier.

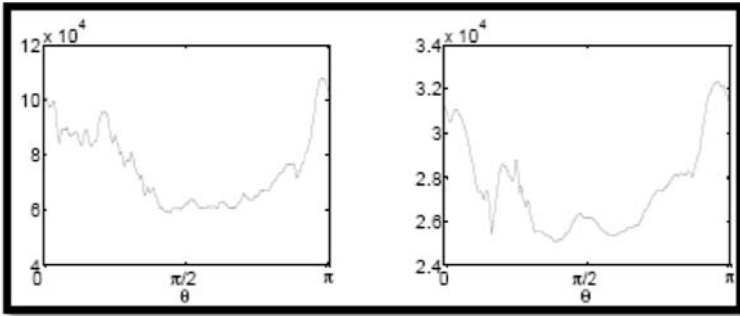


Fig. 4. Circus functions resulting from applying different diametrical functional

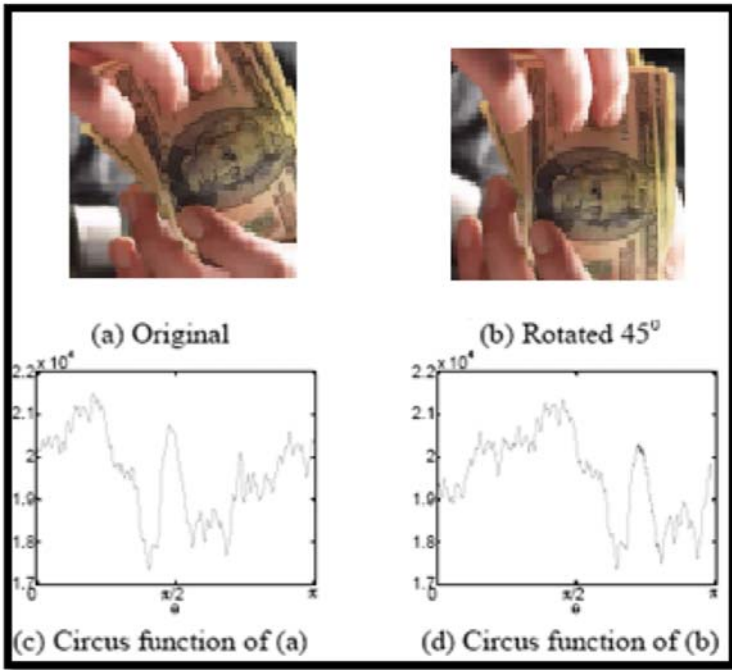


Fig. 5. The circus function (c) for an image (a) and the circus function for the same image rotated by 45° . The circus function is shift to the right by 45° ($\pi/4$).

2.2 Identifier Algorithm

Invariance to shift and amplitude scaling can be achieved by taking the Fourier transform of (5)

$$F(\Phi) = Kexp^{-j\omega\Phi}F[c(a)], \tag{6}$$

and then considering the magnitude of (5)

$$|F(\Phi)| = |KF(\Phi)|, \quad (7)$$

From (7) it can be seen that the original image and the modified image give equivalent descriptors except for the scaling factor K . A binary string is extracted by taking the sign of the difference between neighboring coefficients,

$$b_\omega = \begin{cases} 0 & \text{if } |F(\omega)| - |F(\omega + 1)| < 0 \\ 1 & \text{otherwise} \end{cases} \quad (8)$$

The image identifier is then made up of these values $I = \{i_1, i_2, \dots, i_n\}$ for $n \in N_I$. Results are further improved by using different diametrical functional to extract multiple component identifiers and concatenating them to obtain complete identifier as shown in Figure 6.

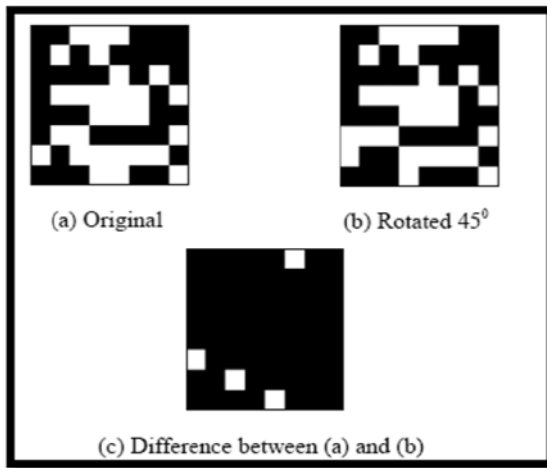


Fig. 6. The binary identifier for an image (a) and its rotated version (b). The difference between the identifiers is shown in (c). The identifier is 1D but has been mapped to 2D for presentation purposes only.

3 Similarity Measure

3.1 The Classical Hausdorff Distance

Given two point sets A and B , the Hausdorff distance[7] between A and B is defined as

$$H(A, B) = \max(h(A, B), h(B, A)), \quad (9)$$

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|, \quad (10)$$

$$h(B, A) = \max_{b \in B} \min_{a \in A} \|a - b\|, \quad (11)$$

where $\|\cdot\|$ denotes some norm of points of A and B . This measure indicates the degree of similarity between two point sets. It can be calculated without an explicit pairing of

points in their respective data sets. The conventional Hausdorff distance, however, is not robust to the presence of noise. Dubuisson et. al. [8] have studied 24 different variations of the Hausdorff distance in the presence of noise. A modified Hausdorff distance (MHD) using an average distance between the points of one set to the other set gives the best result. This measure is the most widely used in the task of object identification and defined as

$$h(B, A) = \frac{1}{n} \sum_{a \in A} \min_{b \in B} \|a - b\|, \tag{12}$$

with $h(B, A)$ defined similarly. This modified Hausdorff distance is less sensitive to noise than the conventional one. It is possible, however, to end show the Hausdorff distance with even more attractive features as it is shown in the next section.

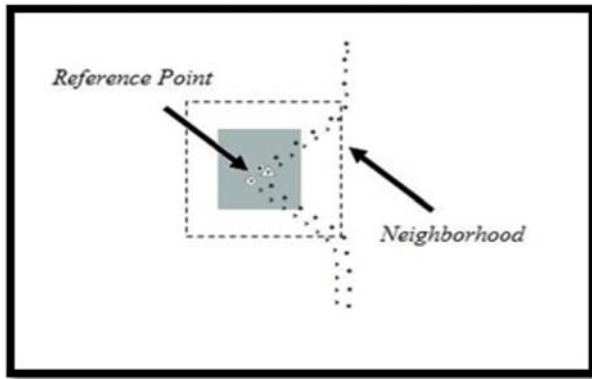


Fig. 7. The grey shade indicates the neighborhood area. The point marked by \circ is a sample point a of the first shape A . The points marked by \blacktriangle and \square are the candidate matching points of the second shape B .

3.2 The Hausdorff- Shape Context

In this section, we propose a shape similarity measure, the “Hausdorff-Shape Context”, based on the combination of the Hausdorff distance and the shape context. The Hausdorff distance measures the distance from point a to all points of set B , $d(a, B)$, then, selects the one at the minimum distance among them. In this case, the candidate point marked by \circ is selected and, then, the distance between them is used as the result. The minimum distance is therefore based only on the spatial information. This is not useful when using the Hausdorff distance in the presence of noise, when we have to deal with the broken point problem caused by segmentation and edge detection errors, etc. To the best of our knowledge, there is no work in the point matching Hausdorff distance with structural point information. We propose an alternative way to find the minimum distance between point a and set B to overcome the above problem. Instead of finding the nearest distance, in our approach, the point descriptor, shape context, is used to find the best matching between point a and set B . We, therefore, call this shape similarity measure as “Hausdorff-Shape context”.

$$h_{HSC}(A, B) = \sum_{a \in A} \omega(a, b') \min_{b \in B} C_{sc}(a, N(b)), \tag{13}$$

and

$$\omega(a, b') = \frac{\mathcal{D}(a, b')}{\sum_{a \in A} \mathcal{D}(a, b')} \text{ and } \sum \omega(a, b') = 1 \tag{14}$$

where $b' = \arg \min_{b \in B} C_{sc}(a, N(b))$ and $C_{sc}()$ is χ^2 test statistic. In the example shown in Fig. 7 the candidate point b' is the one marked by \square which is the correct corresponding point between point a and a point in set B . The cost of matching between two points a and b , $C_{sc}(a, N(b))$ is weighted by their distance, (a, b') . Therefore $\omega()$ is a normalized distance between points a and b' over the entire distance between sets A and B . Furthermore, the neighborhood $N()$ is designed to reduce the computation time of the shape matching, since it finds the best point matching only in the neighborhood area. Thus faster performance improvement can be achieved. The $h_{HSC}(B, A)$ is defined in a similar way. The shape similarity measure in (13) with the maximum Hausdorff matching in (9) is defined to be a confidence level of matching:

$$\text{dist}_{HSC}(A, B) = 1 - H(A, B) \tag{15}$$

4 Experimental Results

The increasing size of image databases, even for consumer applications, means that the false acceptance rate must be kept low to avoid returning large numbers of erroneous matches. To test the performance of the identifier a set of 4,000 original images are used. Each image is modified in 15 different ways to create a dataset of 4,000x16

Table 1. Accuracy rate under different modifications

CONDITION	ACCURACY RATE (%)
Blur 5%	91.89
Blur 10%	92.23
Bright ± 5%	93.03
Bright ± 10%	95.34
Bright ± 15%	91.09
Compression JPEG 95%	99.99
Compression JPEG 80%	99.99
Compression JPEG 65%	97.35
Rotate ±15o	85.43
Rotate ± 30o	85.77
Rotate ±45o	82.87
Scale ±25%	87.67
Scale ±50%	82.98
Scale ±75%	79.57
Flip	85.34
Noise ± 5%	97.55
Noise ± 10%	96.78

images (=64,000). Some example image modifications are shown in Figure 8. All results are presented in terms of the detection rate, which is defined as

$$Acc = 100 \times \frac{a}{A}, \quad (16)$$

where A is the total number of images and a is the number of images correctly identified as matching. In table 1 shows the detection rate results when the false positive and false negative rates are equal



Fig. 8. (a) Original Image (b) with rotate -45° (c) with rotate -45° (d) Bright $+5\%$ (e) Bright $+15\%$ (f) Flip (g) with noise 5% (h) Compression 95%

5 Conclusions

We have presents a robust method for image identification with variant illumination, compression, flip, scaling, rotation and gray scale conversion. Techniques introduced in this work are composed of two stages. First, the signature of image is to be detected by the Trace Transform. Then, in the second stage, the Hausdorff distance and Modified Shape Context are employed to measure and determine of similarity between models and test images. From the experimental result of 60,000 images, the average of accuracy rate is higher than 83%.

Acknowledgement

This research was supported by Mahanakorn University of Technology. The authors would like to thank CSC-MUT for providing testing images.

References

1. Roover, C.D., Vleeschouwer, C.D., Lefèbvre, F., Macq, B.: Robust Video Hashing Based on Radial Projections of Key Frames. *IEEE Trans. on Sig. Proc.* 53(10), 4020–4037 (2005)
2. Ghosh, P., Manjunath, B.S., Ramakrishnan, K.R.: A Compact Image Signature for RTS-Invariant Image Retrieval. In: *Int. Conf. Visual Info. Eng, VIE 2006* (2006)
3. Seo, J.S., Haitisma, J., Kalker, T., Yoo, C.D.: A robust image fingerprinting system using the Radon transform. *Signal Processing: Image Communication* 19, 325–339 (2004)
4. Schmid, C., Mohr, R.: Local Grayvalue Invariants for Image Retrieval. *IEEE Trans. on PAMI* 19(5), 530–535 (1997)
5. Kozat, S., Venkatesan, R., Mihçak, M.K.: Robust Perceptual Image Hashing via Matrix Invariants. In: *IEEE Int. Conf. Image Proc (ICIP 2004)*, pp. 3443–3446 (2004)
6. Kadyrov, A., Petrou, M.: The Trace Transform and Its Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(8), 811–828 (2001)
7. Huttenlocher, P., Klanderman, G., Rucklidge, W.: Comparing Images using the Hausdorff Distance. *IEEE Trans.PAMI* 15(9), 850–863 (1993)
8. Dubuisson, M., Jain, A.K.: A Modified Hausdorff Distance for Object Matching. In: *Proc. ICPR*, pp. 566–568 (1994)
9. Srisuk, S., Fooprateepsiri, R., Waraklang, S.: Object Recognition Robust under Translation, Rotation and Scaling in Application of Image Retrieval. In: *The 7th National Computer Science and Engineering Conference, Burapha University, Chonburi, Thailand, October 28-30*, pp. 249–254 (2003)
10. Srisuk, S., Tamsri, M., Fooprateepsiri, R., Sookavatana, P.: A New Shape Matching Measure for Nonlinear Distorted Object Recognition. In: *International Conference on Digital Image Computing: Techniques and Applications (DICTA 2003)*, Sydney, Australia, December 10-12, pp. 339–348 (2003)

Personalized Fingerprint Segmentation

Xinjian Guo¹, Yilong Yin^{1,*}, and Zhichen Shi²

¹ School of Computer Science and Technology, Shandong University, Jinan 250101, China

² Network Center, Weifang University, Weifang 261061, China

xinjianguo@mail.sdu.edu.cn, ylyin@sdu.edu.cn

Abstract. Fingerprint segmentation is an important pre-processing step in automatic fingerprint identification system. Traditional fingerprint segmentation methods either highly depend on empirical thresholds sophisticatedly chosen by experts or a learned model trained by elements generated from manually segmented fingerprints. It is manpower and time consuming. They always try their best to tune their fingerprint segmentation methods to be universal to all unseen fingerprints. However, one fingerprint may have a significantly distinct distribution from another in feature space because fingerprint acquisition is affected by several factors, such as pressure, the types of sensors, finger tip condition (dry, wet etc.). As a result, the delicate threshold and the well trained model may not be suitable to the new input fingerprints from a new finger or a new person. And it makes worse when automatic fingerprint identification systems meet sensor interoperability. To solve the problem, we propose a personalized fingerprint segmentation method: Automatic Labeling based Linear Neighborhood Propagation (ALLNP), which learns a segmentation model special for each input fingerprint image based on the input image only. The proposed method is tested with typical fingerprint images from four heterogeneous data bases of FVC2000. Experimental results show its effectiveness and encouraging strength when fingerprint segmentation meets sensor interoperability.

Keywords: Fingerprint recognition, Fingerprint segmentation, Semi-supervised learning, Label propagation, Linear Neighborhood Propagation.

1 Introduction

Owing to uniqueness and immutability of fingerprint [1], it has been used as one of the biometrics features for a very long time. An automatic fingerprint identification system (AFIS) consists of several steps, such as fingerprint segmentation, image enhancement and filtering, binarization, thinning, gaining minutiae of fingerprint matching, and so on. Fingerprint segmentation is important as a pre-processing step in AFIS. A captured fingerprint image mainly consists of two components: foreground and background. The foreground is the component that originates from the contact of the fingertip with the sensor, and the background is the noisy area at the border of the image. The purpose of fingerprint segmentation is separating foreground of high quality from background and

* Corresponding author.

foreground of low quality or unrecoverable. Effective fingerprint segmentation not only decreases the computational cost in the subsequent steps but also improves the system performance.

Fingerprint segmentation typically extracts features (or single feature) for every element first, which can be a pixel or an un-overlapped block of the input fingerprint image. Then what segmentation methods need to do is to decide the type (foreground or background) of each element. Statistical features of grey level, e.g., mean and variance of pixel intensity, directional image, ridge projection signal and Gaussian-Hermite moments are often used in fingerprint segmentation. Mehre [2] proposed a segmentation method based on directional image. To overcome the limitations of [2] when the input image has perfectly uniform regions, a composite segmentation method [3] is suggested using the variance criterion wherever the directional method fails. Bazen [4] proposes a completely different solution based on pixel-wise direction and coherence. Bazen [5] trains an optimal linear classifier based on three pixel features: coherence, mean and variance (CMV). Yin [6] trains a quadric surface model based on pixel-wise CMV features. Ratha [7] computes the variance of the projection signal on different directions with the prior knowledge that the foreground block is of large variance along the direction orthogonal to the ridges and is of small variance along the direction parallel to the ridges, and background is usually of small variance along all directions. Wang [8] proposes to segment fingerprint based on Gaussian-Hermite moments. Jain [9] takes texture energy of each pixel and their spatial locations as input to a squared-error clustering algorithm. Helfroush [10] proposes a modified method based on Jain [9], but uses dominant ridge score of each block instead of coherence, and takes median filtering as a post processing step to improve the performance of the fingerprint segmentation. Yin [11] proposes a segmentation method consisting of two steps: in the primary segmentation, non-ridge regions and unrecoverable low quality ridge regions are removed as background by a well trained neural network, and the secondary segmentation, the remaining ridges are identified and removed according to the two typical differences between the remaining ridges and the true ridges. Bernard [12] proposes a multiscale Gabor wavelet filter bank using the Phase of Multiscale Gabor Wavelets for a robust and efficient fingerprint segmentation. Ross [13] apply convex hull algorithm to Fingerprint segmentation. Klein [14] uses a hidden Markov model (HMM) to solve the problem of fragmented segmentation.

Although there are lots of researches on fingerprint segmentation, they either highly depend on empirical thresholds sophisticatedly chosen by experts or a learned model trained by samples generated from manually segmented fingerprints. It is manpower and time consuming. They always try their best to tune their fingerprint segmentation methods to be universal to all unseen fingerprints. However, one fingerprint may have a significantly distinct distribution from another in the feature space, as shown in Section 2, because fingerprint acquisition is affected by several factors, such as pressure, the types of sensors, finger tip condition (dry, wet etc.). As a result, the delicate thresholds and the well trained models may not be suitable to the new input fingerprint from a new finger or a new person. And it makes worse when fingerprint verification meets sensor interoperability [15]. To the best of our knowledge, there is no research on how to segment a fingerprint image based on the input fingerprint image only. Thus we argue

that personalized fingerprint segmentation makes more sense. Here, *personalized* means fingerprint segmentation result for one fingerprint relies only on the input fingerprint image. The contribution of the paper is two folds. For one, to realize personalized fingerprint segmentation, we propose Automatic Labeling based Linear Neighborhood Propagation (ALLNP) method, which learns from the input fingerprint image only instead of a set of fingerprints, and segments the input fingerprint image specifically. For another, to avoid fragmented blocks in segmented fingerprints to some extent, we take position information of elements, i.e., block row index and block column index in the paper, as new segmentation features. Experiments show encouraging strength of the proposed method in sensor interoperability.

The remainder of the paper is organized as follows. Section 2 presents a new formulation of fingerprint segmentation in transductive view. Our method ALLNP is proposed in section 3. Section 4 contains the experimental results. And section 5 concludes the paper.

2 Formulation of Fingerprint Segmentation in Transductive View

Traditional fingerprint segmentation methods are analyzed in this section theoretically and empirically, followed by a new formulation formulated in the paper. As we stated in Section 1, almost each of previous fingerprint segmentation methods either chooses an empirical threshold sophisticatedly according to experience of experts or learns a model by samples generating from manually segmented fingerprints by experts. However, it is unreasonable to learn a fingerprint segmentation model in such a way, especially when the fingerprint images, on which the model is trained, have distinct distribution in feature space. For instance, they are captured via sensors of different types.

Fig.1 shows the scenario when traditional fingerprint segmentation methods do not work. For one input fingerprint image denoted by elliptic dots, Hyperplane1 can easily separate it. And for another input fingerprint image, Hyperplane1 can easily separate it. However, when a segmentation model is trained by a mixture of samples generated from the two input fingerprint images, it seems difficult to find an exact hyperplane suitable for the two and subsequent numerous input fingerprints.

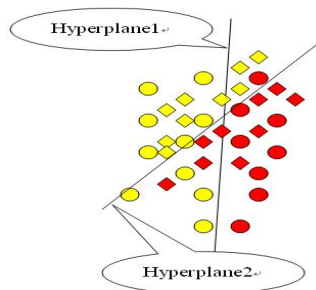


Fig. 1. Illustration of traditional fingerprint methods. Elliptic dots represent samples (elements) from one fingerprint image, while diamondoid ones represent samples from another fingerprint image. Dots in yellow color represent foreground samples, while ones in red color represent background samples.

Formally, we assume the input fingerprint image can be divided into n un-overlapped blocks denoted by $\mathcal{X} = \{x_1, x_2, \dots, x_l, x_{l+1}, \dots, x_n\}$, where $x_i \in R^d$, and let $\{y_1, y_2, \dots, y_l, y_{l+1}, \dots, y_n\}$ represents the class (foreground or background) of blocks in the fingerprint image, where $y_i \in \{1, -1\}$, 1 for foreground and -1 for background. The fingerprint segmentation task is to learn a hypothesis $f \in F$. And it is unsupervised learning. However, if we can get some prior knowledge of what blocks are most likely foreground and background ones. In other word, if we can get the first l labels $\{y_1, y_2, \dots, y_l\}$ corresponding to $\{x_1, x_2, \dots, x_l\}$, we can transfer the knowledge from the labeled data to unlabeled data. The learning task will become a transductive learning [16], since we only anticipate its generalization ability on a definite and closed data set. In the next section, we exhibit an oracle how to partially label blocks in a fingerprint image.

To validate the rationality of the new formulation in fingerprint segmentation preliminarily, we select two typical fingerprints from NIST-4 [17], and project them to CMV space the most commonly used in fingerprint segmentation, as shown in Fig.2. The original fingerprint images are listed in the top left, and their histograms in individual dimension of CMV space are aligned in the top right and the second row correspondingly. It can be seen that the two fingerprints have significantly different distribution in CMV feature space.

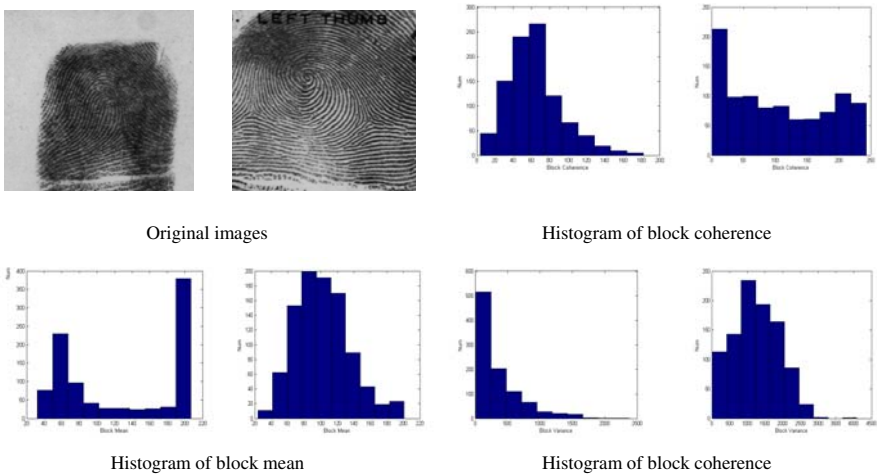


Fig. 2. Distribution of typical fingerprint images in CMV space

3 The Proposed Method: ALLNP

In the section, an Automatic Labeling based Linear Neighborhood Propagation (ALLNP) is proposed, which in fact is a self-help semi-supervised fingerprint segmentation method.

Before conducting semi-supervised learning on the dataset generated from the input image, an oracle is used to label some unlabeled data automatically. ALLNP works as in Fig. 3. An input fingerprint image to be segmented is first divided into un-overlapped blocks¹. Then a feature vector is extracted for each block. Subsequently, some definitely foreground and background blocks are automatically by an oracle. Provided with these labeled data (blocks) L and the remaining unlabeled data U in the image, a graph-based semi-supervised method called linear neighborhood propagation (LNP) is adopted to do the transductive learning on D , resulting in the segmented fingerprint. Some readers may be confused and argue why we conduct an oracle to label only some data points instead of all. Selectively labeling some data points is a much easier task than labeling all, so we take the easier task as a mediate step of the more complex task.

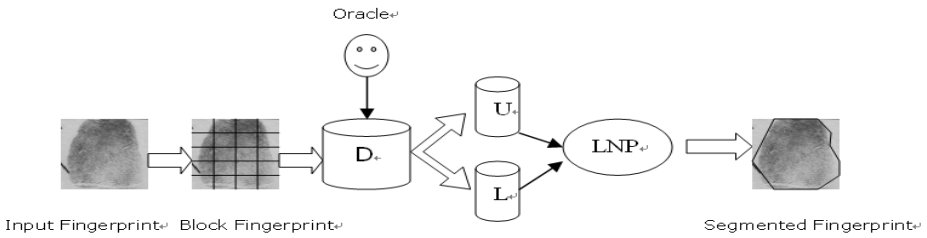


Fig. 3. Flow diagram of the proposed method ALLNP

3.1 Block Contrast as an Oracle

We have investigated several commonly used image features. And block contrast seems to be a more discriminative feature. Suppose an input fingerprint image is divided into a set of $w \times w$ blocks. For one block B , block contrast is defined to be the quotient of block variance and block mean, as shown

$$Block\ Contrast_B = \frac{Block\ Variance_B}{Block\ Mean_B} \tag{1}$$

The block mean for block B is defined to be

$$Block\ Mean_B = \frac{1}{w \times w} \sum_{(x,y) \in B} I_{(x,y)} \tag{2}$$

where $I_{(x,y)}$ is the intensity of the pixel (x, y) . And the block variance for block B is defined as the variance of intensity of each pixel in the block B , represented by .

¹ In the paper, we segment fingerprint images in a block-wise way.

$$Block\ Variance_B = \frac{1}{W \times W} \sum_{(x,y) \in B} (I_{(x,y)} - Block\ Mean_B)^2 \quad (3)$$

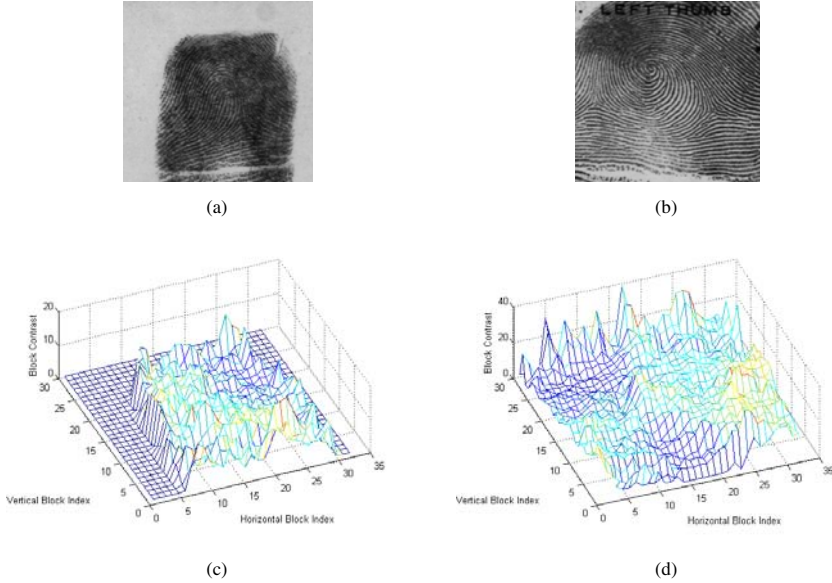


Fig. 4. The example plots of block contrast

Fig.4 shows two example plots of block contrast for two fingerprints, where both images are represented by 16×16 blocks. (a) and (b) in Fig.4 represent the two fingerprints, while (c) and (d) are their plots of block contrast respectively. In the two block contrast plots, x-axis and y-axis represent block indices in horizontal and vertical directions of the original fingerprint images respectively, and the z-axis stands for block contrast value of each block.

In the paper, block contrast is taken as an oracle to automatically label some foreground and background blocks for an input image. For each block, block contrast, as defined in (1), is extracted first. Then, each block is sorted into a list according to its block contrast in ascending order. Blocks in the top of the list have larger probabilities to be background blocks than those in the bottom, while these in the bottom have larger probabilities to be foreground ones than those in the top.

3.2 Label Propagation by LNP Algorithm

The graph-based semi-supervised learning methods have received considerable attraction in recent years, which model the whole dataset as a graph. The construction of the graph is at the heart of these graph-based methods. And most of these methods [18, 19] adopt a Gaussian function to calculate the edge weights of the graph but the variance of

the Gaussian function will affect the classification results significantly. To address the above limitation of graph-based semi-supervised learning, Linear Neighborhood Propagation [20] is proposed.

The reason why we select LNP as our solution is twofold. First, the number of the nearest neighbors k in LNP is easier to tune since it is selected from only positive integers in a small range. Some other semi-supervised learners, such as S3VM and co-training, need to explicitly specify the ratio of two classes², or implicitly assume the unlabeled data has the same ratio with labeled data. That may be improper for fingerprint segmentation problem. Because fingerprint images captured by different sensors actually have different proportions of foreground owing to various resolutions of sensors. The proportions of foreground for the same finger acquired by the same sensor may distinctly differ if with different pressures. Every fingerprint can be seen as a manifold embedded in a high space. The parameter k may be inherently affected by fingerprints, and insensitive. Second, LNP has been shown of the capability to automatically erase the noise in labeled data. So even we injected some noise in the automatic labeling the first step of our algorithm, LNP still works.

The LNP algorithm consists of two steps. In the first step, it approximates the whole graph by a series of overlapped linear neighborhood patches, and the edge weights in each patch can be constructed by solving the following standard quadratic programming problem

$$\begin{aligned} \min_{w_{i,j}} & \sum_{j,k:x_j,x_k \in N(x_i)} w_{ij} G_{jk}^i w_{ik} \\ \text{s.t.} & \sum_j w_{ij} = 1, w_{ij} \geq 0 \end{aligned} \tag{4}$$

Where $N(x_i)$ represents the neighborhoods of x_i , w_{ij} is the contribution of x_j to x_i , and G_{jk}^i represents the (j, k) -th entry of the local Gram matrix $(\mathbf{G})_{j,k} = (x_i - x_j)^T (x_i - x_k)$ at point x_i , where $(\cdot)_{j,k}$ represents the (j, k) -th entry of a matrix. Then all the edge weights will be aggregated together to form the weight matrix of the whole graph. In the second step, the labels of the labeled data to the remaining unlabeled data using the constructed graph in the first step. In detail, (5) is used to update the labels of each data object until convergence.

$$f^{t+1} = \alpha \mathbf{W}f^t + (1 - \alpha)y \tag{5}$$

Where $0 < \alpha < 1$ is the fraction of label information that x_i receives from its neighborhoods. Let $y = (y_1, y_2, \dots, y_n)^T$ with $y_i = L_i (i \leq l)$, $y_u = 0 (l + 1 \leq u \leq n)$, $f^t = (f_1^t, f_2^t, \dots, f_n^t)^T$ is the prediction label vector at

² Unless otherwise specified, the paper talks about two class problem.

iteration t and $f^0 = y$. And LNP has been derived from a regularization framework to provide a theoretical guarantee of its feasibility [20].

4 Experiments

In this section, we present some experimental results of our personalized fingerprint segmentation algorithm ALLNP. In order to validate the strength of ALLNP in segmenting fingerprints of sensor interoperability, it is tested with typical fingerprints from 4 heterogeneous databases of the Fingerprint Verification Competition 2000 (FVC2000) [21]. For the reason that the fingerprint segmentation result needs human inspection, we select 10 typical fingerprints of different quality from each fingerprint database. So there are 40 images in all in our test set. We divided each input fingerprint image into a set of 16 by 16 blocks, then a feature vector consisting of block mean, block variance, block contrast and block coherence is extracted for each block. Besides, to avoid fragmented blocks in the segmented fingerprints to some extent, we take position feature of each block, i.e., block row index and block column index, as new features. In all the experiments, the parameter α is set to 0.99, which stands for the fraction of label information that a block receives from its neighbors in feature space in each iteration. The number of neighbors calculated for each block seems insensitive in fingerprint segmentation, and it is set to be 7 in the experiment for all input fingerprint images.

Some segmentation results by our method without any post-processing are shown in the Fig.5. Two fingerprints are selected from each data set. Images in the first column are input fingerprints. And the second column shows corresponding partially automatically labeled fingerprints of the first column. For each input fingerprint 20 foreground and 10 background blocks are automatically labeled. To distinguish the automatically labeled foreground and background blocks we deal with them as follows. Labeled foreground blocks are displayed the same intensity as these in the input image, and labeled foreground ones are displayed as black, while unlabeled blocks are displayed as white. Some fingerprints in the second column have black margin, because the sizes of their input images can not divide by the block size. And we simply segment the margin to be background. Segmented fingerprints by ALLNP are shown in the last column. It can be seen that the proposed personalized fingerprint method ALLNP achieves favorable segmentation results on almost all the fingerprints, which indicates its strength in sensor interoperability.

Some statistical experiment results of previous fingerprint segmentation methods available are listed in Table.1 for comparison. It is worth noting that these figures were quoted simply from their papers, and we did not realize these methods. It can be observed that our method ALLNP is better than all the other methods except Yin 2005 [6]. With human inspection our personalized fingerprint segmentation method ALLNP achieves an encouraging fingerprint segmentation performance with an error rate of only 2.89% in block-wise segmentation. And post-processing will reduce the error rate of our method further.

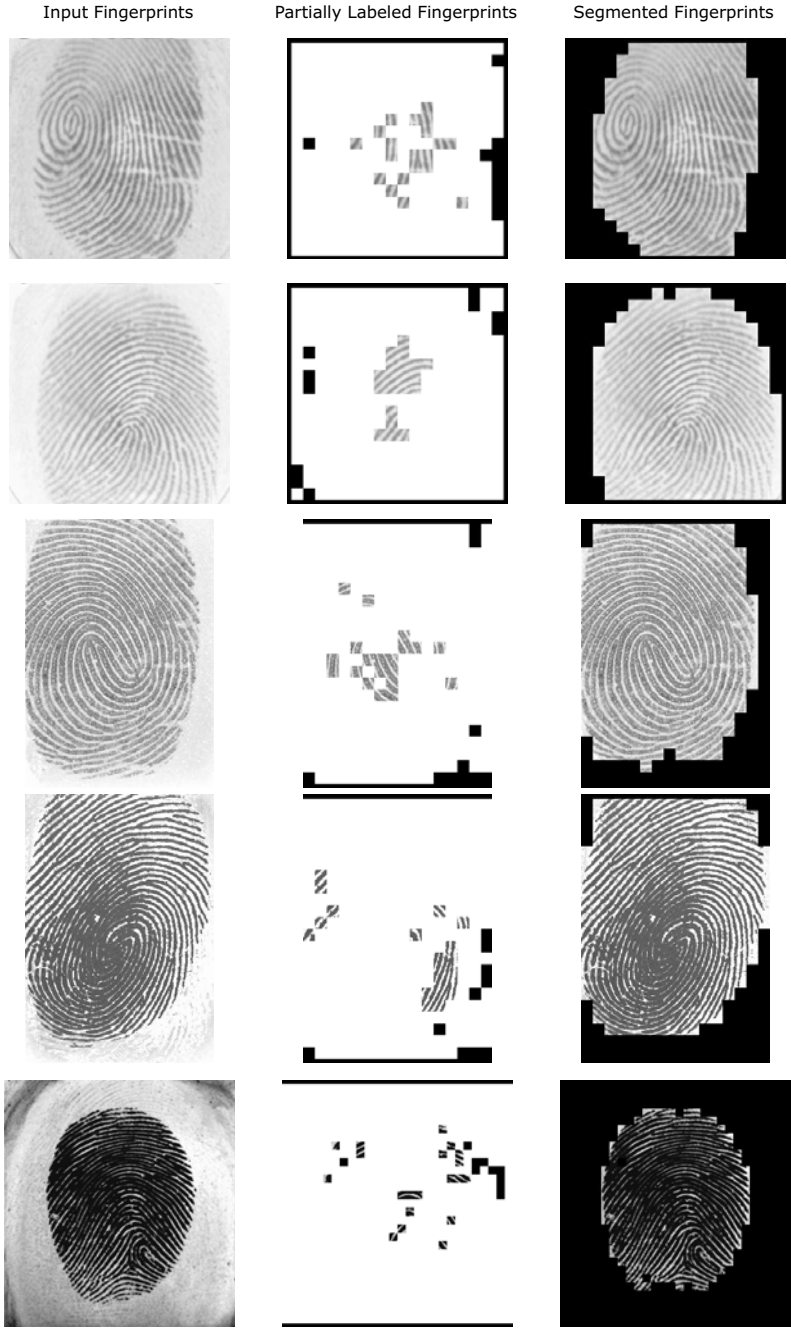


Fig. 5. Segmentation results of ALLNP on some typical fingerprints of FVC2000

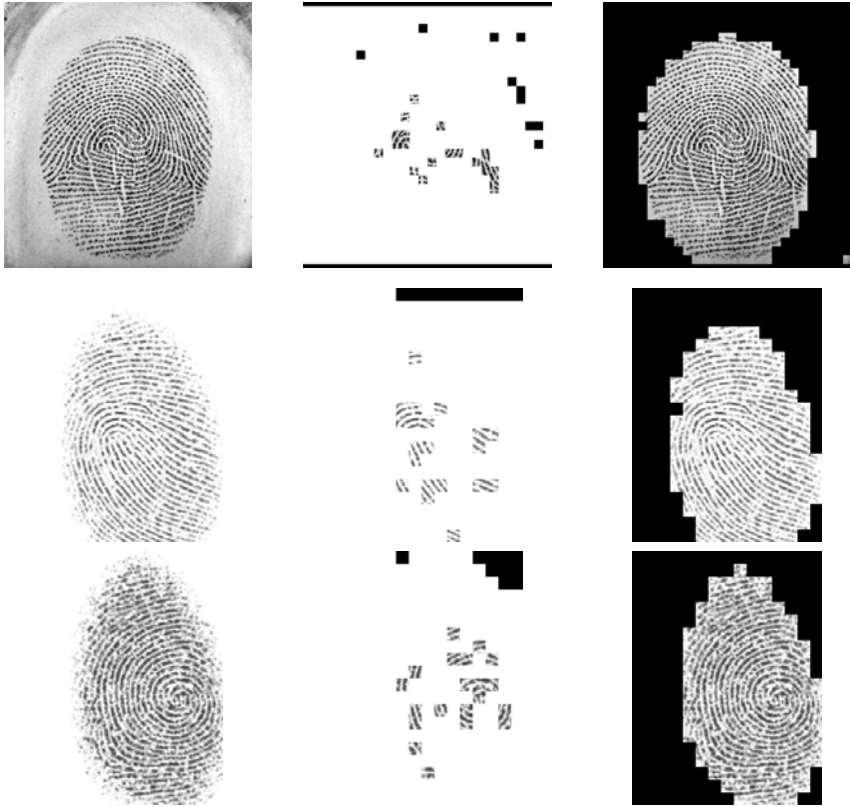


Fig. 5. (Continued)

Table 1. Comparison of fingerprint segmentation methods. In the third and the fourth columns, “Y” denotes yes, “N” denotes no, while “-” represents unknown from the paper.

Methods	Block-wise/Pixel-wise	Pre-processing	Post-processing	Error Rate
Bazen 2001 [5]	Pixel-wise	Y	Y	6.8%
Klein 2002 [14]	Block-wise	-	N	6.5%
Yin 2007 [11]	Block-wise	-	-	>10.6%
Yin 2005 [6]	Pixel-wise	-	Y	0.53%
Bernard 2002 [12]	Pixel-wise	Y	-	> 1.85%
ALLNP	Block-wise	N	N	2.89%

Although our algorithm has achieved above advantage, it is worthy to mention that almost all the experiments of previous fingerprint segmentation methods are carried out on single homogeneous fingerprint data set, in which all the fingerprints are obtained

via the same sensor. When the trained models by these fingerprint segmentation methods are tested on several heterogeneous fingerprint data sets, their performance will significantly drop.

5 Conclusion

Traditional fingerprint segmentation methods always try their best to tune their fingerprint segmentation methods to be universal to all unseen fingerprints. However, one fingerprint may have a significantly distinct distribution from another in the feature space because fingerprint acquisition is affected by several factors. As a result, the delicate threshold and the well trained model may not be suitable to the new input fingerprints from a new finger or a new person. And it makes worse when automatic fingerprint identification systems meet sensor interoperability. In the paper, we propose a personalized fingerprint segmentation method ALLNP, which learns a fingerprint segmentation model specially for an input fingerprint image based on the input image only. The proposed method is tested with representative fingerprints from four heterogeneous databases of FVC2000. The experiments show encouraging performance of the proposed method when fingerprint segmentation meets sensor interoperability. However, in Section 3, some foreground and background blocks are automatically labeled by a simple oracle based on block contrast before learning. And the block numbers automatically labeled for each input fingerprint are small. We may wish more labeled data, for the more exactly labeled data provided the better segmentation performance it achieves. However, some noise may be injected as the number of automatically labeling blocks increases. In the future work, we will investigate robust automatically labeling mechanics.

Acknowledgments. The authors thank Liming Zhang and Yanbing Ning in the MLA Group for their great help in revising the paper. The work is supported by Shandong Province High Technology Independent Innovation Project under Grant No. 2007ZCB01030 and Shandong Province Natural Science Foundation under Grant No. Z2008G05.

References

1. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer, New York (2003)
2. Mehtre, B.M., Murthy, N.N., Kapoor, S.: Segmentation of fingerprint images using the directional image, *J. Pattern Recognition* 20(4), 429–435 (1987)
3. Mehtre, B.M.: Segmentation of fingerprint images – a composite method, *J. Pattern Recognition* 22(4), 381–385 (1989)
4. Bazen, A.M., Gerez, S.H.: Directional field computation for fingerprints based on the principal component analysis of local gradients. In: Proceedings of the ProRISC 2000, 11th Annual Workshop on Circuits, Systems and Signal Processing, Veldhoven, The Netherlands (November 2000)

5. Bazen, A.M., Gerez, S.H.: Segmentation of fingerprint images. In: Proceedings of Workshop on Circuits Systems and Signal Processing (ProRISC 2001), pp. 276–280 (2001)
6. Yin, Y.L., Wang, Y.R., Yang, X.K.: Fingerprint image segmentation based on quadric surface model. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 647–655. Springer, Heidelberg (2005)
7. Ratha, N., Chen, S., Jain, A.K.: Adaptive flow orientation-based feature extraction in fingerprint images, *J. Pattern Recognition* 28(11), 1657–1672 (1995)
8. Wang, L., Dai, M., Geng, G.H.: Fingerprint image segmentation by energy of Gaussian-Hermite moments. In: Li, S.Z., Lai, J.-H., Tan, T., Feng, G.-C., Wang, Y. (eds.) SINOBIOOMETRICS 2004. LNCS, vol. 3338, pp. 414–423. Springer, Heidelberg (2004)
9. Jain, A.K., Ratha, N.K.: Object detection using Gabor filters. *J. Pattern Recognition* 30(2), 295–309 (1997)
10. Helfroush, M.S., Mohammadpour, M.: Fingerprint Segmentation. In: Proceedings of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA 2008), pp. 1–5 (2008)
11. Yin, J.P., Zhu, E., Yang, X.J., Zhang, G.M., Hu, C.F.: Two steps for fingerprint segmentation. *J. Image and Vision Computing* 25(9), 1391–1403 (2007)
12. Bernard, S., Boujemaa, N., Vitale, D., Bricot, C.: Fingerprint Segmentation Using the Phase of Multiscale Gabor Wavelets. In: The 5th Asian Conference on Computer Vision, Melbourne, Australia (January 2002)
13. Ross, A.: Information Fusion in Fingerprint Authentication, Ph.D. Thesis, Michigan State University (2003)
14. Klein, S., Bazen, A., Veldhuis, R.: Fingerprint image segmentation based on hidden Markov models. In: Proceedings of 13th Annual Workshop on Circuits, Systems, and Signal Processing, vol. 2002, pp. 310–318 (2002)
15. Ross, A., Jain, A.K.: Biometric Sensor Interoperability: A Case Study In Fingerprints. In: Maltoni, D., Jain, A.K. (eds.) BioAW 2004. LNCS, vol. 3087, pp. 134–145. Springer, Heidelberg (2004)
16. Zhu, X.J.: Semi-supervised learning literature survey. Technical Report 1530, Department of Computer Sciences, University of Wisconsin, Madison (2005)
17. Watson, C.I., Wilson, C.L.: NIST Special Database 4, Fingerprint Database. National Institute of Standards and Technology (March 1992), <http://www.nist.gov/srd/nistsd4.htm>
18. Szummer, M., Jaakkola, T.: Partially Labeled Classification with Markov Random Walks. *Advances in Neural Information Processing Systems* 14 (2002)
19. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with Local and Global Consistency. *Advances in Neural Information Processing Systems* 16 (2004)
20. Wang, F., Wang, J.D., Zhang, C.S., Shen, H.C.: Semi-Supervised Classification Using Linear Neighborhood Propagation. In: Proceeding of 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2006), June, 2006, New York University, New York (2006)
21. Maio, D., Maltoni, D., Cappelli, R., Wayman, J.L., Jain, A.K.: FVC2000: Fingerprint Verification Competition. In: ICPR, Barcelona (September 2000), <http://www.bias.csr.unibo.it/fvc2000>

Automatic Image Restoration Based on Tensor Voting

Toan Nguyen, Jonghyun Park, Soohyung Kim, Hyukro Park, and Gueesang Lee

Department of Electronics and Computer Engineering, Chonnam National University
toan_mulmi@hotmail.com, gslee@chonnam.ac.kr

Abstract. An automatic image restoration method is proposed for text images despite severe occlusion and noise. 3D tensor voting framework is used to analyze surface areas to detect corrupted regions. These corrupted regions are then restored by an adaptive median filter or image completing. The experimental results attained from several text images show that good images can be achieved from degraded ones by using the proposed method.

Keywords: tensor voting, restoration, image repairing, image completion.

1 Introduction

Some portions of natural scene images can be corrupted due to occlusion or noises such as dusts, streaks, shadows and small unwanted objects. Several methods can be used to restore these corrupted regions such as variational image inpainting [2] and image completion [9]-[11]. Corrupted regions in the input images of these methods, however, are located and marked manually by users. In this paper, we propose a novel method to detect and locate the corrupted regions automatically by using 3D tensor voting. Whereas large corrupted regions can be successfully corrected by the image completion, in the paper we only focus on small and medium noise sizes.

Image restoration has been developed in three distinct fields: variational image inpainting, texture synthesis and image completion. Narrow gaps or corrupted regions can be successfully filled by variational image inpainting methods that are based on prolonging the isophotes arriving at the boundary. These methods exploit the continuity of the geometrical structure of an image to fill the corrupted regions. The most important work, a partial differential equation (PDE)-based algorithm, was presented in [2]. Although the whole image looks fine with PDE-based variational image inpainting methods, the details are blurring because textured images cannot be filled and continuation is not well enough. Some other methods in variational image inpainting are level lines [1], detecting edges [3] and global approach [4].

For real images with large corrupted regions, texture information is important. Many texture synthesis methods have been reported in the literature. The most popular texture synthesis method is based on statistical model [5]. In this method, the authors modeled and matched the statistical features of the sample texture. This method, however, captures only marginal statistics and joint properties between different scales and orientations are not considered. Other approaches are image-based [6] and patch-based [7], which generate good results for many applications. Variational image inpainting and texture synthesis are combined in the image completion. The first work in the

image completion was presented in [8]. In this method, an input image is decomposed into its structure and texture components. Image inpainting and texture synthesis are then applied to these components separately. The result is obtained by adding the two processed components together. The method exploits the advances of two methods image inpainting and texture synthesis but it is slow and gives blurry outputs due to diffusion. Another image completion method is fragment-based [9]. In [9], surrounding information of pixels is determined by a confidence map. Based on the confidence map, the color of an unknown region is inferred from visible parts of the image. More confident pixels are considered first. In each step, unknown region is filled by a similar fragment that is found. This method gives good result but it is extremely slow and complex. A simpler and faster than fragment-based method is exemplar-based method [10]. In video, a space-time video completion is proposed in [11]. The missing areas are filled by sampling spatio-temporal patches from available parts of the video.

The work in [12], a tensor voting-based image segmentation method, is the closest to our work. Since the surface saliency values of tokens, image pixels, are directly proportional to the areas of regions they belong to, the small corrupted regions are detected based on the surface saliency map. In this method, the input tokens of tensor voting are generated from a chromaticity image with value range is from 0 to 1. Since the chromaticity image contains real values, pixels on the same surface are not well aligned together. Therefore, the surface saliency map is not clean to infer noise regions, especially in low quality natural images. In our method, the k-means clustering is applied on the input image to separate objects into different layers. The corrupted regions, therefore, are easily detected. An adaptive median filter or image completion method is applied on corrupted regions to recover the original image.

The remainder of the paper is organized as follows. The 3D tensor voting framework is reviewed in section 2. Our automatic image restoration method and results are presented in sections 3 and 4, respectively. Section 5 gives conclusions and draws future work.

2 Tensor Presentation and Voting in 3D

Tensor voting (TV) [13], [14] is a unified computational framework to solve a wide range of computer vision problems. In our application, 3D tensor voting is used to analyze 3D surface in order to detect corrupted regions. Each pixel in the input image is represented by a triple $(x, y, H(x, y))$ representing for row, column, and a value achieved from a function of color or grayscale information of this pixel. Each pixel, or token, belongs to a geometric structure such as region, curve, surface or the intersection among them. To extract geometric structure information, each token is represented by a second order tensor. Each tensor is represented by a 3 by 3 matrix and visualized as an ellipsoid whose shape indicates the type of structure presented and its size the saliency of this information. An isolate pixel can be represented by a ball tensor that indicates a structure which has no preference of orientation. A plate tensor can represent for a token corresponding to a curve element with curve tangent vector is the smallest eigenvector. A stick tensor indicates an elementary surface token with the biggest eigenvector is its normal. Here, however, we do not know in advance what type of entity a token may belong to. Furthermore, since features may overlap, a location may actually correspond to several types at the same time. Therefore, a token may

be represented by a generic tensor that can be decomposed into stick tensor, plate tensor and ball tensor which correspond to the three terms in the following equation.

$$\begin{aligned}
 T &= \lambda_1 \hat{e}_1 \hat{e}_1^T + \lambda_2 \hat{e}_2 \hat{e}_2^T + \lambda_3 \hat{e}_3 \hat{e}_3^T \\
 &= (\lambda_1 - \lambda_2) \hat{e}_1 \hat{e}_1^T + (\lambda_2 - \lambda_3) (\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T) \\
 &\quad + \lambda_3 (\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T + \hat{e}_3 \hat{e}_3^T),
 \end{aligned}
 \tag{1}$$

where λ_i are the eigenvalues in decreasing order and \hat{e}_i are the corresponding eigenvectors. After encoding the input tokens by tensors with initial values, these tensors communicate with each other in order to derive the most preferred orientation information for each of the input tokens, and extrapolate the inferred information at every location in the domain for the purpose of coherent feature extraction. In other words, each tensor votes its neighboring tensors with its information and also receives votes from them. The shape and size of this neighborhood and the vote strength and orientation are encapsulated in predefined voting fields or kernels. The voting field for each tensor component is used to look up the orientation and magnitude of the votes cast. All voting fields are based on the fundamental 2-D stick voting kernel (Fig.1). The orientation of the stick vote is normal to the smoothest circular path connecting from the voter to the recipient. The magnitude of the vote is calculated by the following decay function.

$$DK(s, k, \sigma) = e^{-\left(\frac{s^2 + ck^2}{\sigma^2}\right)},
 \tag{2}$$

where $s=(l\theta)/\sin(\theta)$ and $k=2\sin(\theta)/l$. The parameter s is the arc length from the voter to the recipient, k is the curvature, c is a constant, and σ is the scale of voting field controlling the size of the voting neighborhood and the strength of votes. Note that the vote strength at Q' and Q'' is smaller than at Q because Q' is father and Q'' requires higher curvature than Q . Each token in the domain receives several votes from its neighboring tokens. Vote accumulation is performed by tensor addition or equivalently by addition of 3 by 3 matrices. After voting, saliency maps of each kind of tensor are computed. To analyze surface saliency of input tokens, we build the surface saliency map calculate from value $(\lambda_1 - \lambda_2)$ of each resulting token.

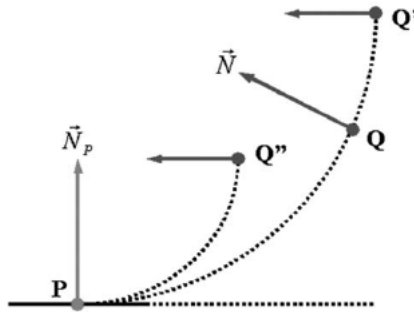


Fig. 1. Vote generation by a 2D stick tensor

3 Image Restoration Based on 3D TV

The flowchart of our method is illustrated by Fig.2. The k-means clustering method is applied on the input image to generate a segmented image. The segmented image, which contains objects in several layers, is used as input data for the 3D tensor voting framework. The surface saliency map achieved from 3D tensor voting is analyzed to detect corrupted regions. An adaptive median filter or an image completion technique is then applied on the original image with corrupted regions marked by a noise map. If all corrupted regions are not successfully recovered, the enhanced image is fed to the system again for next iterations.

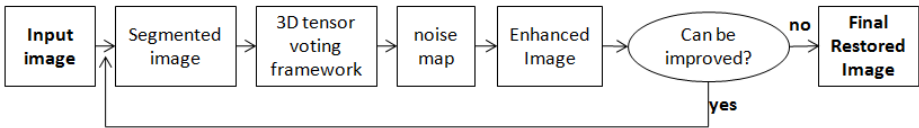


Fig. 2. The flowchart of the proposed method

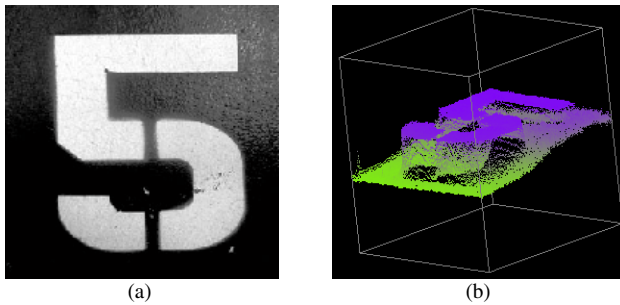


Fig. 3. Chromaticity image with method in [12], (a): original image, (b): chromaticity image

3.1 Generating Input Data

Since the tensor voting framework is a general tool for solving many computer vision problems, input data should be represented correctly for a specific application. To analyze and detect corrupted regions appearing as small regions in the image, different objects or surfaces should be presented separately in different surfaces in 3D spaces of tensor voting domain. In [12], a chromaticity image is created based on values of pixels in the input image. Since the value in the chromaticity image is real, pixels that belong to different objects or layers are not well separated. Fig.3 shows an example of a chromaticity image created by the method in [12]. Some parts of the image are not easily classified to background or foreground. To enhance this preprocessing step, we propose to use the k-means clustering with color information on the input image to create separate layers for different objects.

The k-means clustering is a well-known method of cluster analysis [15]. If the input image is a grey scale image, a 2-means clustering with grey scale value or global thresholding binarization can be used. For a color input image, in the ideal case, the

number of clusters is same as the number of dominant colors in the image. Since our method is iterative, we do not need to detect and correct all noise regions at once. Therefore, k can be set to 3 or 4. To reduce the effect of uneven illumination on clustering result, we convert input color image from RGB to $L^*a^*b^*$ color space and apply k-means clustering on a^* and b^* color components.

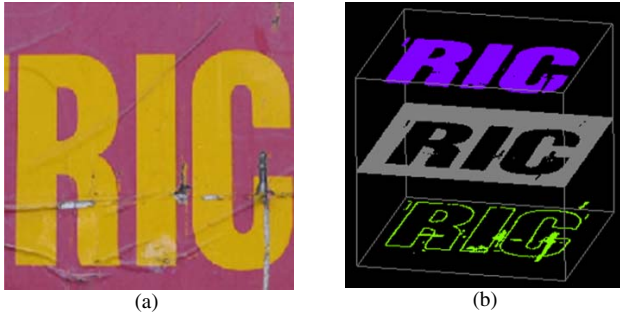


Fig. 4. Segmented image generated by k-means clustering, (a): original image, (b): clustered image with 3-means clustering method

Fig.4 shows an example of clustering the input image into 3 clusters. With k-means clustering, objects with different colors are well separated to different layers. Note that each layer may contain many objects such as background and noise at the same time. By applying tensor voting, we intend to take out the small and isolate regions considered as noise or corrupted regions.

3.2 Detecting Corrupted Regions

The segmented image is then used as token data. Each token corresponding to a pixel at (x, y) is represented by a triple $(x, y, H(x, y))$ where $H(x, y)$ is its cluster index. Tokens are encoded by tensors and communicate each other in voting process. The value of the scale of voting field is calculated based on experiments ($\sigma=10$ with 256×256 images). The surface saliency map is calculated as the magnitude of the biggest eigenvector of stick component of accumulated tensors. Color coding presentation of saliency maps of image in Fig.3 are depicted in Fig.5. Tensors that lie on smooth salient features (i.e., curves or surfaces) strongly support each other. For this reason, tokens belonging to large surface areas have larger saliency values than that of tokens belonging to small surface areas. Since corrupted regions have small areas, they can be detected based on surface saliency values. With our method, the surface saliency map is very flat and small saliency areas are easily detected. The noise map contains all tokens whose surface saliency values are lower than a threshold.

3.3 Recovering Corrupted Regions

The noise map that is used as marked regions in image inpainting indicates corrupted regions that should be restored. An image completion method can be used on these corrupted regions. Since in our application the corrupted regions are small, an adaptive median filter also can be used. The adaptive median filter applying on each channel of the original image is presented as following algorithm.

Algorithm 1. ADAPTIVE MEDIAN FILTER

For each noise pixel (x, y) in the noise map:

STEP 1: Collect all pixels that are not marked in the noise map of a considering channel of the original image in the $(m \times n)$ window surrounding the noise pixel. The number of pixels is N .

STEP 2: Test if the number of pixels is enough.
If $(N < TH)$ then $m = m + 2, n = n + 2$ and GOTO STEP 1.
Else GOTO STEP 3

STEP 3: Assign median value among the enumerated values for pixel at (x, y) .

The improved version in current iteration is compared with the improved version in previous iteration. If the different between them is not much, the algorithm stops. Otherwise, it is fed into a new iteration.

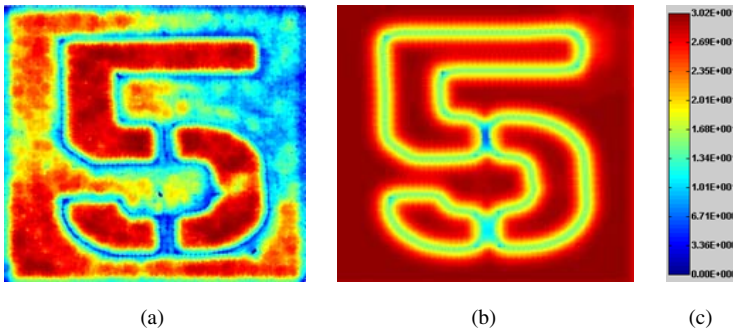


Fig. 5. Surface saliency map, (a) using method in [12], (b) using our proposed method, (c): saliency value for color code

4 Experimental Results

Several color text images are selected to evaluate the performance of the proposed method. Some examples are represented in Fig. 6. The original images are shown in the second column. After detecting noise region by tensor voting, the restored images by adaptive median filter and image completion [11] are depicted in the third and the last columns, respectively. The first column shows the index of the current iteration. With a few iterations (less than 3), all small noises are gone and the enhanced version of the original image is achieved. These enhanced images are now ready for next processing steps such as binarization. The image completion method [11] gives a better result compared to the simple adaptive median filter, especially in the boundary of the image. The image completion also remains good texture information of the corrupted regions. This method, however, very complex compared to the adaptive median filter. It takes several minutes to complete recovering noises in a 256×256 image. In the first

image of Fig. 6, the number of dominant colors is 4 but with 3-means clustering method we can correct some noises in the first iteration. Remaining errors in improved version are completely corrected in the second iteration. Our proposed method converges in a fewer number of iterations compared to the method in [12].

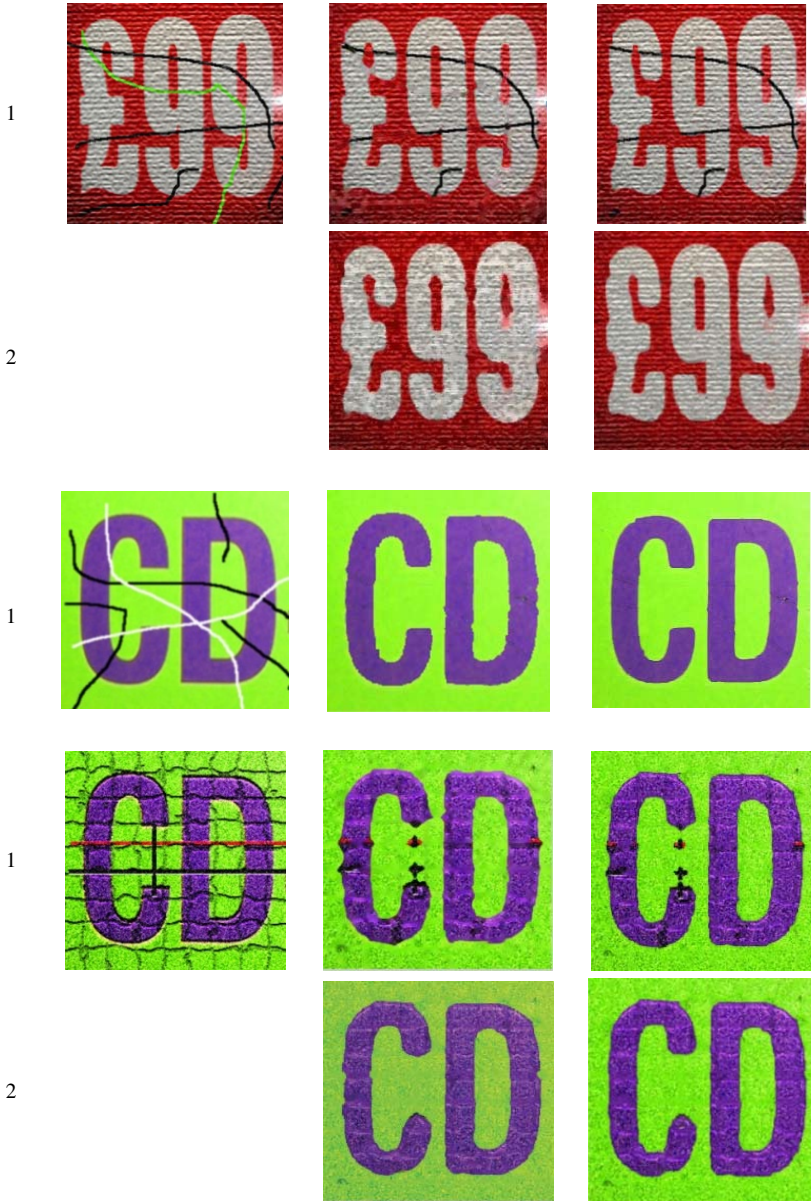


Fig. 6. Some automatic image restoration results, (a): iteration index, (b): original images, (c): results by adaptive median filter, (d): results by image completion [11]



Fig. 6. (continued)

5 Conclusions

In this paper, an automatic image restoration for text images having small noise regions is proposed based on 3D tensor voting. The k-means clustering method is used to create input data for the tensor voting framework. By analyzing the surface saliency map, small regions considered as noise regions are detected. After the noise regions are located correctly, an adaptive median filter or image completion method can be applied to recover corrupted regions by using information of neighboring pixels. The experimental results show that our method can generate good results for many complex text images.

Acknowledgement

This work was supported by the Korea Research Foundation Grant funded by the Korean Government(KRF-2008-313-D00999) and the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency (NIPA-2009-C1090-0903-0008)).

References

1. Masnou, S., Morel, J.: Level lines based disocclusion. In: Proc. Of International Conference on Image Processing (1998)
2. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: SIGGRAPH 2000: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp. 417–424. ACM Press/Addison-Wesley Publishing Co., New York (2000)

3. Rares, A., Reinders, M.J.T., Biemond, J.: Edge-based image restoration. *IEEE Transactions on Image Processing* 14(10), 1454–1468 (2005)
4. Auclair-Fortier, D.Z.M.-F.: A global approach for solving evolutive heat transfer for image denoising and inpainting. *IEEE Transactions on Image Processing* 15(9), 2558–2574 (2006)
5. Heeger, D.J., Bergen, J.R.: Pyramid-based texture analysis/synthesis. In: *SIGGRAPH*, pp. 229–238 (1995)
6. Efros, A., Leung, T.: Texture synthesis by non-parametric ampling. In: *ICCV*, vol. (2), pp. 1033–1038 (1999)
7. Lin, W.-C., Hays, J., Wu, C., Liu, Y., Kwatra, V.: Quantitative evaluation of near regular texture synthesis algorithms. In: *CVPR 2006: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 427–434 (2006)
8. Bertalmio, M., Vese, L., Sapiro, G., Osher, S.: Simultaneous tructure and texture image inpainting. In: *CVPR*, vol. 02, p. 707 (2003)
9. Drori, I., Cohen-Or, D., Yeshurun, H.: Fragment-based image completion. In: *SIGGRAPH 2003*, pp. 303–312. *ACM Press*, New York (2003)
10. Criminisi, P.P.A., Toyama, K.: Region filling and object removal by exemplar-based inpainting. *IEEE Trans. Image Processing* 13(9), 1200–1212 (2004)
11. Wexler, Y., Shechtman, E., Irani, M.: Space-Time Video Completion. In: *Computer Vision and Pattern Recognition (CVPR)*, Washington (2004)
12. Lim, J., Park, J., Medioni, G.G.: Text segmentation in color images using tensor voting. *Image and Vision Computing* 25(5), 671–685 (2007)
13. Guy, G., Medioni, G.: Inference of Surfaces, 3D Curves, and Junctions from Sparse, Noisy, 3-D Data. *IEEE Trans. on PAMI* 19(11), 1265–1277 (1997)
14. Medioni, G., Lee, M.S., Tang, C.K.: *A Computational Framework for Segmentation and Grouping*. Elsevier, Amsterdam (2000)
15. Hartigan, J.A.: *Clustering Algorithms*. Wiley, Chichester (1975)

Robust Incremental Subspace Learning for Object Tracking

Gang Yu, Zhiwei Hu, and Hongtao Lu

MOE-Microsoft Laboratory for Intelligent Computing and Intelligent Systems,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, 200240, China
{skicy,huzhiwei,htlu}@sjtu.edu.cn

Abstract. In this paper, we introduce a novel incremental subspace based object tracking algorithm. The two major contributions of our work are the Robust PCA based occlusion handling scheme and revised incremental PCA algorithm. The occlusion handling scheme fully makes use of the merits of Robust PCA and achieves promising results in occlusion, clutter, noisy and other complex situations for the object tracking task. Besides, the introduction of incremental PCA facilitates the subspace updating process and possesses several benefits compared with traditional R-SVD based updating methods. The experiments show that our proposed algorithm is efficient and effective to cope with common object tracking tasks, especially with strong robustness due to the introduction of Robust PCA.

1 Introduction

During the past decades, object tracking is rapidly developed since it is widely used in many different areas, like surveillance, human computer interface, enhanced reality and so forth. The accuracy and robustness of object tracking influence the performance of these applications. Therefore, object tracking has been a hot research area in computer vision and a lot of researches have been explored in this area. The main challenge of object tracking is the difficulty in handling the appearance variability of a target object. There are two categories of appearance variabilities, intrinsic and extrinsic appearance variabilities. The intrinsic appearance variabilities mainly include shape deformation and pose variation of a target object. On the other hand, changes in illumination, changes in viewpoint and partial occlusion belong to extrinsic variabilities. All the appearance variabilities pose great challenges to accurately locate the target object including the well-known methods [1,2]. However, subspace based methods, solving this problem by modeling such appearance variabilities in low-dimension space, prove to be efficient and effective in [3,4]. [3] first brought eigenspace to model appearance changes of target objects. The advantages of this subspace representation are several folds. Firstly, the subspace representation provides a compact notion of "thing" to be tracked rather than "stuff", which means structure information is fully utilized in the appearance representation. Besides, this

method also survives in large appearance changes. But the needs to train the appearance model before starting the program and solve complex optimization problems limit the use of this method. Later, Lim et al improves this method in [4] by using R-SVD [7] to incrementally update the subspace and Particle Filter to replace the complex optimization steps. Due to the merits of stochastic methods, local minimum problem caused by deterministic optimization methods is well solved. Based on [3], [15] makes use of Rao-Blackwellized Particle Filter, achieving promising results in clutter environment. Lin et al [16] further optimizes the framework of [4] according to the idea of Fisher Discriminant Analysis(FDA). The import of the second subspace makes the method more discriminative since the utilization of background appearance. Meantime, Ho et al [11] replaces the traditional L^2 reconstruction error norm with uniform L^2 reconstruction error norm and achieves promising experimental results. Recently, Zhang et al [12] utilizes the framework of Graph Embedding and proposes a new discriminative subspace representation. Besides, Log-Euclidean Riemannian Subspace [14] and Tensor Subspace [13] are also brought in to handle the appearance variabilities. Although the theory parts and experimental results of these methods sound attractive, the overall framework is almost the same and similar with [4]. The possible differences lie on the subspace representation and corresponding R-SVD based updating algorithm. The import of Log-Euclidean and Tensor subspace strengthen the robustness and accuracy of tracking results. However, in the meantime, they also add additional complexities to the tracking framework, and the tracking speed may be influenced by the complicated subspace representations. Accordingly, the disadvantages may limit the wide use of these methods. Our method, on the other hand, avoids the complexity of elaborately subspace representation and adopts the traditional PCA-based representation. To obtain robust and accurate tracking performance, a Robust PCA [5] is utilized in our framework.

Two components are inevitable in subspace based methods. One is the subspace representation and the other is the algorithm to update the subspace incrementally. Although different subspaces are utilized to model the appearance variabilities, almost all the above methods update their corresponding subspace based on the R-SVD algorithm. In this paper, however, we will adopt a new incremental subspace updating algorithm, which possesses several beneficial advantages compared with R-SVD. Furthermore, according to the experiments of previous subspace methods, although occlusion may be handled well in the simple situations, the performance is deteriorated when the scenes become complex. On the other hand, if the subspace is updated when occlusion happens, outliers will bias the subspace and probably make the tracking results drift from the target region. In order to cope with these problems, a novel occlusion handling scheme is proposed in our paper. The main idea of this scheme is based on Robust PCA.

The rest of paper is structured as follows. Section 2 describes our subspace representation. Updating scheme(Incremental PCA) and our algorithm framework

are discussed in section 3. Section 4 gives some experimental results of our method and section 5 concludes this paper.

2 Robust PCA Based Learning

Principle Component Analysis is one of the traditional dimension reduction methods which has been widely used in computer vision group. Since PCA minimizes L^2 reconstruction error, it is also considered as one of most successful reconstructive methods. By projecting a new sample into a pretrained subspace, the reconstruction error can be regarded as a useful tip for deciding whether the new sample is a kind of object that is similar with the training set. Hence, the intrinsic nature of PCA makes it practical in object tracking area. There are numerous works dedicated to make use of the merits of PCA in the object tracking programs. Some of fundamental and influential works are [3,4]. Although many works try to further improve the discriminability of subspace based methods, robustness is actually one of the essential problems currently which limits the wide application of subspace based methods. According to our experimentations, it can be easily found that the tracking methods lose their targets not because of lacking discriminative abilities but because of lacking robustness, especially in complex situations that occlusion and fast movement of target object happen. Thus, our method is proposed not to strengthen the discriminability of subspace based methods but to increase the robustness of PCA based methods.

2.1 Robust PCA

Let n be the number of images in the training set, each of which having m pixels. The training data set then can be represented by $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$, $\mathbf{x}_i \in \mathbb{R}^m$, where \mathbf{x}_i refers to the i th training image. We will use the notation $U, U \in \mathbb{R}^{m \times k}$ for the truncated eigen basis where k means the number of bases we keep.

For the traditional PCA, $U = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k]$, $\mathbf{u}_i \in \mathbb{R}^m$ is calculated by minimizing the reconstruction error:

$$E = \sum_{i=1}^n \left(\mathbf{x}_i - \sum_{j=1}^k a_{ij} \mathbf{u}_j \right)^2 \quad (1)$$

where $a_{ij} = \mathbf{u}_j^T \mathbf{x}_i$. To solve Eq. (1), either Eigen Decomposition or SVD(Singular Value Decomposition) can be used. The goal of reconstructive methods is to find \mathbf{a}_i once a new sample arrives. In traditional PCA, $\mathbf{a}_i = U^T \mathbf{x}_i$. However, when there are outliers or noises in the image \mathbf{x}_i , the coefficients \mathbf{a}_i may be influenced by these contaminated pixels. Robust PCA, on the other hand, can limit the influences of outliers and noises and achieve robust results.

In the following part, a brief discussion of Robust PCA will be presented. For the detail information, we can refer to [5]. To achieve robustness, subsampling is employed in the calculation of coefficients \mathbf{a} . The full process can be reviewed

as a hypothesize-and-select paradigm using only subsets of image pixels. There are two major steps, generating hypotheses and selection.

First of all, suppose U is calculated from a training data set, let us return to Eq. 1. Due to only subsets of pixels of a new image sample being considered, we need to seek the solution of \mathbf{a} which minimizes

$$E(\mathbf{r}) = \sum_{i=1}^q \left(x_{r_i} - \sum_{j=1}^k a_j(\mathbf{x})u_{j,r_i} \right)^2 \tag{2}$$

where $\mathbf{r} = [r_1, r_2, \dots, r_q]$, $k < q < m$ refers to q points selected from m pixels in a new image \mathbf{x} .

The minimization of Eq. 2 can be easily solved by least square. Then, in the first step of Robust PCA, several hypotheses are generated, each one referring to a subset of points(\mathbf{r}). For each hypothesis, in each step of minimization, we get one temporary solution of coefficients \mathbf{a} and the corresponding reconstruction error for each point($\xi_i = x_i - \sum_{j=1}^k a_j u_{ji}$). Through trimming part of the points whose ξ_i are above a threshold, a new solution of \mathbf{a} can be obtained with the trimmed set of points. This iterative step continues until the number of points in the hypothesis is below a predefined threshold. In the final hypothesis, a notion of compatible points is defined as follow:

$$D = \{j|\xi_j^2 < \theta\}, \quad \text{where } \theta = \frac{2}{m} \sum_{i=k+1}^n \lambda_i \quad (\lambda_i \text{ is eigen value}) \tag{3}$$

The cardinality of the compatible points set is denoted as $s = |D|$ which can provide useful information for the selection step.

According to this method, several candidate hypotheses for \mathbf{a} (each \mathbf{a} represents a potential coefficients vector computed from a subset of sample points with Eq. 2) are generated. The optimal one is selected to maximize the following function:

$$c_i = K_1 s_i - K_2 \|\xi\|_{D_i}, \quad \text{where } \|\xi\|_{D_i} = \sum_{j \in D_i} \xi_j^2$$

where s_i and $\|\xi\|_{D_i}$ refer to the number of compatible points and the reconstruction error over the set D_i (D_i refers to a set of pixels from one image, from which the coefficients \mathbf{a} is computed), and the coefficients K_1 and K_2 are parameters.

2.2 Occlusion Handle Scheme

In this subsection, a carefully designed occlusion handling scheme will be discussed. Though the scheme is mainly applied to deal with the occlusion situation, it is useful for some complex situations like out-of-plane rotation and clutter background as well.

For the tracked object, there will be three possible states for each frame. One is the normal state, meaning that all the conditions are normal and there is

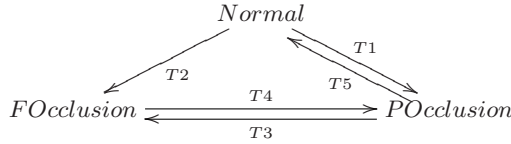


Fig. 1. State Transition Graph

no special arrangement for this state. The second kind of state is partial occlusion(POcclusion), which is mainly used for preventing the false updating of the subspace when occlusion happens. The last state is full occlusion(FOcclusion), in which we need to increase the particle number and state variance in order to relocate the target object. Besides, since the target may be fully occluded, we do not update the target position with the new estimation. For simplicity, we denote t as the frame number in the video. Fig. 1 is a simple description of the possible transitions for the three states.

At time frame t , the five transitions happen when certain requirements are met.

$$\begin{array}{lll}
 T1: \gamma_t > \theta_1 & T2: \gamma_t > k\theta_1 & T3: \gamma_t > k\theta_1 \\
 T4: \gamma_t \leq k\theta_1 & T5: \delta_t < \theta_2 &
 \end{array}$$

If none of these requirements are met, it means that the target keeps still in the original state. In the above requirements, θ_1, θ_2 are thresholds to decide whether current state is occluded, and $k(k > 1)$ is a coefficient. γ_t and δ_t refer to the reconstruction difference based on the robust coefficients(\mathbf{a}_t) and summation of these differences. They can be calculated as follows:

$$\delta_t = \sum_{t-\theta_3 < j \leq t} \alpha^{t-j} \gamma_j \qquad \gamma_t = \|\mathbf{x}_t - U_t \mathbf{a}_t\|_2 \tag{4}$$

where θ_3 is the number of frames to consider and α is a forget factor($\alpha = 0.9$ in our experiments).

Intuitively, we can easily interpret these five transitions as below. When no occlusion happens, the robust reconstructed image will differ little from the original image, which means that the requirement of T1 cannot be met. However, once occlusion happens, γ_t will be certainly larger than θ_1 and the target will fall to the partial occlusion state, which means the subspace should not be updated due to the noises and outliers in the new image samples. In the same time, we set a higher threshold $k\theta_1$ for indicator of full occlusion. When full occlusion happens, we keep the target object still in the last position until the full occlusion state is stopped. Besides, in order to locate the target object when the target appears again, the variance of state variable and number of particles are increased. Once we find a state meeting the requirement of T4, the variance of state variable and number of particles return to the original values. If the requirement of T5, the latest summation of reconstruction differences is below the threshold θ_2 , is met, it means that the target is no longer in occlusion state and the updating step can be started again.



Fig. 2. Tracking results based on occlusion handling scheme

We illustrate the state transitions in Fig. 2. There are two examples in Fig. 2. In the first row, the first and fifth images refer to the normal state. The second and fourth images represent the partial occlusion state. Full occlusion state is illustrated in third image. There are two subimages in each image, representing the target object in the frame and reconstructed target object with robust coefficients. It is obvious that updating with the reconstructed target object can prevent noises and outliers from biasing the subspace. The second row shows another successful example of our occlusion handling scheme.

We illustrate the state transitions in Fig. 2 with two examples. There are two subimages in each image, representing the target object in the frame and reconstructed target object with robust coefficients. It is obvious that updating with the reconstructed target object can prevent noises and outliers from biasing the subspace.

3 Proposed Tracking Algorithm

3.1 Overview of the Approach

The framework of our method is similar with [4]. There are two major components of the framework, locating the target region and updating the subspace. The visual tracking problem can be formulated as an inference problem based on Hidden Markov Model, where X_t and I_t refer to hidden state variable(target region) and observed variable(video frame) respectively. Let $X_t = (x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t)$, where $x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t$ denote x translation, y translation, rotation angle, scale, aspect ratio and skew direction at time t . According to Bayesian theorem, we have:

$$p(X_t|I_t) \propto p(I_t|X_t) \int p(X_t|X_{t-1})p(X_{t-1}|I_{t-1})dX_{t-1} \quad (5)$$

Due to the difficulties in directly calculating the posterior probability $p(X_t|I_t)$, stochastic approximation methods like particle filter [1] are adopted to approximate the probability with a stochastically generated set of weighted samples. The dynamic model($p(X_t|X_{t-1})$) is modeled by a Gaussian distribution:

$$p(X_t|X_{t-1}) = N(X_t; X_{t-1}, \psi) \quad (6)$$

To introduce probabilistic interpretation, the observation model is modeled with PPCA [10], which is widely used in subspace based object tracking methods. For simplicity, the observation model in our method is governed by a Gaussian distribution:

$$p(I_t|X_t) = N(I_t; \mu, UU^T + \epsilon I) \propto \exp(-\|(I_t - \mu) - U\mathbf{a}_t\|^2) \tag{7}$$

where I is an identity matrix, ϵ is the additive Gaussian noise in the observation process, μ is the mean of training images and \mathbf{a}_t is the coefficients of X_t outputted by Robust PCA. The detail proof can refer to [8].

3.2 Incremental PCA

Once a new target location is estimated, the subspace need to adapt to the new appearance change unless the target is predicted as occlusion in our occlusion handling scheme. Although almost all of previous works adopt the R-SVD [7] based incremental methods, we try to use a different incremental scheme(IPCA) [6]. The benefits are several folds. The first is of course the computation efficiency. Also, IPCA is more extendable and flexible due to the possibility to integrate spatial and temporal weights. Furthermore, IPCA can perfectly integrated with Robust PCA and our occlusion scheme and do not require any training images of the target object before the tracking task starts. The IPCA algorithm can be viewed in Algorithm. 1. For convenience, we suppose $t > k$ in our algorithm framework. For $t \leq k$, it is also easy to deduce from the algorithm.

Algorithm 1. Incremental PCA

Input: Subspace eigen vectors $U_t \in \mathbb{R}^{m \times k}$, eigen value $D_t \in \mathbb{R}^k$, coefficients $A_t \in \mathbb{R}^{k \times t}$, mean vector $\mu_t \in \mathbb{R}^m$, new input image $\mathbf{x}_{t+1} \in \mathbb{R}^m$

Output: $U_{t+1}, D_{t+1}, A_{t+1}, \mu_{t+1}$

1. Get the Robust coefficients of \mathbf{x}_{t+1} on current subspace: $\mathbf{a} = \text{RobustPCA}(U_t, D_t, \mu_t)$ with Eq. 2
 2. Reconstruct the image: $\mathbf{y} = U_t\mathbf{a} + \mu_t$
 3. Calculate reconstruction error $\mathbf{r} \in \mathbb{R}^m : \mathbf{r} = \mathbf{x}_{t+1} - \mathbf{y}$
 4. Form new basis vector: $U' = [U_t \quad \frac{\mathbf{r}}{\|\mathbf{r}\|}]$
 5. Determine coefficients in new basis: $A' = \begin{bmatrix} A'_t & \mathbf{a} \\ \mathbf{0} & \|\mathbf{r}\| \end{bmatrix}$
- $$A'_t = \begin{cases} A_t & \text{if } t < \theta_4 \\ A_t(:, 2 : t) & \text{if } t \geq \theta_4 \end{cases}$$
6. Perform PCA on A' obtaining mean value μ'' , eigenvectors U'' and D_{t+1} , discard the last part of columns of U'' and denotes it as $U^* = U''(:, 1 : k)$.
 7. $A_{t+1} = U^{*T}(A' - \mu''\mathbf{1})$, $U_{t+1} = U'U^*$, $\mu_{t+1} = \mu_t + U'\mu''$
-

Since previous target information are well preserved in sample coefficients(A_t), the calculation of subspace bases do not depend on the storage of previous

samples($\mathbf{x}_i, i = 1, \dots, t$). This greatly reduces the number of memory storage. Besides, in order to reduce the impact of earliest frames and increase the influence of latest frames, the earliest frame will be omitted if the number of frame coefficients we keep is above a threshold(θ_4). Also, the updating sample is not the original one which may contain noises and outliers. We use the images reconstructed based robust PCA coefficients. This is feasible only in IPCA framework in which most of the operations are based on sample coefficients.

3.3 Summary of Our Tracking Algorithm

The two major components, target location estimation and online updating scheme, are seamlessly embedded into our algorithm framework with occlusion handling scheme to increase the robustness of our algorithm. To get a general idea of how our method works, a summary of our tracking algorithm is depicted in Algorithm 2. The first three steps in Algorithm 2 are similar with traditional particle filter based algorithms except the introduction of the results of Robust PCA in Eq. 7. The addition of the final two steps increase the robustness of our algorithms with the help of occlusion handling scheme.

Algorithm 2. Summary of Proposed Algorithm

For each frame I_t :

1. Generate particle set $\{x_t^{(i)}\}_{i=1:N}$ with dynamic model $p(X_t|X_{t-1})$ (Eq. 6)
 2. Compute the weight of each particle with Eq. 7
 3. Find the particle with largest weight, marked it as x_t^{opt}
 4. Decide the target state according to occlusion handling scheme and execute corresponding measures (Section 2.2)
 5. If the target stays in Normal state, update the subspace with IPCA(Algorithm 1)
-

4 Experimental Results

Numerous videos have been tested for our proposed algorithms. Due to the limitation of paper length, only a compelling example is illustrated here(Fig. 3). The first row shows the results of our proposed algorithm. In order to illustrate the state transition of our method, we draw the particle information in the second row. When full occlusion happens in the fourth and fifth images, the number of particles and the variance of particle state variables are both increased. Once the target is relocated again, these variables return to normal values showed in the sixth image. The third row shows the tracking results of ISL, which fails when occlusion happens. Some of the quantitative results are also given in the following table, in which the first row shows the average location error(pixels) of ISL and the second row is the result of our method. The video and ground truth files are downloaded from [9].

faceocc	faceocc2	coke11	sylv	tiger1	tiger2
42.5432	35.8175	31.1371	16.3283	40.3083	53.6643
12.0424	19.3204	27.7316	15.8316	34.5523	49.6800



Fig. 3. Tracking results of our robust method (the first row) and ISL (the third row). The second row shows the particle information of our method. The frame number is 106, 115, 117, 119, 126, 129.

5 Conclusion

We have presented a robust incremental subspace based object tracking algorithm whose efficiency and robustness can be found out in our experiments. The two major contributions of our method are the occlusion handling scheme and the revised incremental PCA algorithm. With the help of Robust PCA, the occlusion handling scheme contributes a lot to the robustness of our method, which not only successfully solve the occlusion problem but also can improve the tracking results in noisy and clutter scenes. On the other hand, instead of using the traditional R-SVD based updating methods, the incremental PCA algorithm gives more flexibility and efficiency to our method.

Although the experiments show promising results for our method, there are also several shortcomings needing to improve. The tracking speed is still a common problem related with subspace based tracking algorithms. Besides, our method may fail when the target object experiences fast out-of-plane movements or large light variance. We aim to address these issues in our future works.

Acknowledgement

This work was supported by the Open Project Program of the National Laboratory of Pattern Recognition (NLPR), 863 Program of China (No. 2008AA02Z310) and NSFC (No. 60873133).

References

1. Isard, M., Blake, A.: Condensation: conditional density propagation for visual tracking. *International Journal of Computer Vision* 1, 5–28 (1998)
2. Avidan, S.: Ensemble tracking. In: *Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 494–501 (2005)
3. Black, M.J., Jepson, A.D.: Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision* 26, 63–84 (1998)
4. Lim, J., Ross, D., Lin, R.S., Yang, M.H.: Incremental learning for visual tracking. *Advances in Neural Information Processing Systems* 1, 793–800 (2004)
5. Leonardis, A., Bischof, H.: Robust recognition using eigenimages. *Computer Vision and Image Understanding* 78, 99–118 (2000)
6. Skocaj, D., Leonardis, A.: Weighted and robust incremental method for subspace learning. In: *International Conference on Computer Vision*, vol. 2, pp. 1494–1501 (2003)
7. Levy, A., Lindenbaum, M.: Sequential Karhunen-Loeve Basis Extraction and its Application to Images. *IEEE Transactions on Image processing* 9, 1371–1374 (2000)
8. Ross, D.A., Lim, J., Lin, R.-s., Yang, M.-h., Lim, J., Yang, M.-h.: Incremental Learning for Robust Visual Tracking. *International Journal of Computer Vision* 77, 125–141 (2008)
9. http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml
10. Tipping, M.E., Bishop, C.M.: Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society* 61, 611–622 (1999)
11. Ho, J., Lee, K.C., Yang, M.H., Kriegman, D.: Visual Tracking Using Learned Linear Subspaces. In: *Computer Vision and Pattern Recognition*, pp. 782–789 (2004)
12. Zhang, X., Hu, W., Maybank, S., Li, X.: Graph Based Discriminative Learning for Robust and Efficient Object Tracking. In: *International Conference on Computer Vision* (2007)
13. Li, X., Hu, W., Zhang, Z., Zhang, X., Luo, G.: Robust Visual Tracking Based on Incremental Tensor Subspace Learning. In: *International Conference on Computer Vision* (2007)
14. Li, X., Hu, W., Zhang, Z., Zhang, X., Zhu, M., Cheng, J.: Visual Tracking Via Incremental Log-Euclidean Riemannian Subspace Learning. In: *CVPR* (2008)
15. Khan, Z., Balch, T., Dellaert, F.: A rao-blackwellized particle filter for eigentracking. In: *CVPR*, pp. 980–986 (2004)
16. Lin, R.-s., Ross, D., Lim, J., Yang, M.-h.: Adaptive discriminative generative model and its applications. In: *Advances in Neural Information Processing Systems*, pp. 801–808 (2004)

Reversible Data Hiding Using the Histogram Modification of Block Image

Hyang-Mi Yoo¹, Sang-Kwang Lee², Young-Ho Suh², and Jae-Won Suh¹

¹ Chungbuk National University, College of Electrical and Computer Engineering,
12 Gaeshin-dong, Heungduk-gu, Chongju, Korea
hmYoo82@cbnu.ac.kr, sjwon@cbnu.ac.kr

² Electronics and Telecommunications Research Institute,
161 Gajeong-Dong, Yuseong-Gu, Daejeon, Korea
sklee@etri.re.kr, syh@etri.re.kr

Abstract. Reversible data hiding has drawn considerable attention in recent years. Reversible data hiding recover the original image without any distortion after the hidden data have been extracted. However, one of drawbacks for existing reversible data hiding is underflow and overflow. To overcome these problems, we propose a new reversible data hiding algorithm based on histogram modification of block image. The experimental results and performance comparisons with other reversible data hiding schemes are presented to demonstrate the validity of our proposed algorithm.

Keywords: Reversible Data Hiding, Histogram Shift, Underflow and Overflow, Block Image, Hash Code.

1 Introduction

Multimedia contents can be easily and widely distributed by the illegal copy, which is serious to contents owners. Data hiding technique can be a good solution to protect copyright of the contents by embedding secret information. In recent years, reversible data hiding had been studied vigorously for sensitive image authentication, such as military image and medical image. Reversibility means that original image is completely recovered from embedded image without distortion after embedded message has been extracted.

There are some reversible data hiding algorithms. Fridrich *et al.* [1] losslessly compresses some selected bit plane(s) to leave space for data embedding. Difference expansion scheme by Tian [2] selects some expandable difference values of pixels, and embed one bit into each of them. However, location map should be embedded with payload data to know which difference values have been selected for the difference expansion. Alattar [3][4] has extended Tian's work by generalizing the difference expansion technique for any integer transform.

Recently, some reversible data hiding algorithms based on histogram modification have been reported in the literature. Ni *et al.* [5][6] utilizes the zero or the minimum points of histogram of an image and slightly modifies the pixel

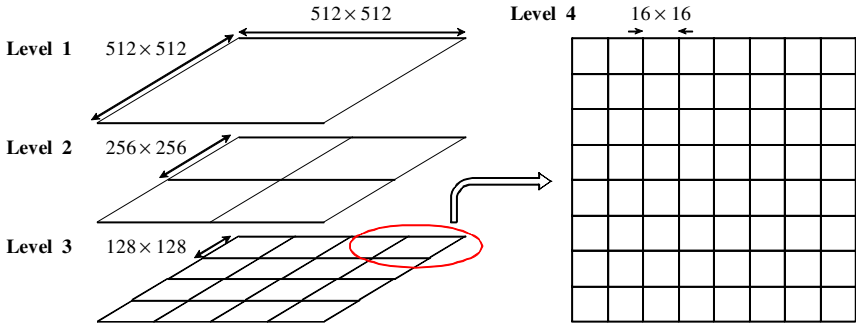


Fig. 1. Partitioning of an Image and Four Level Hierarchical Block Structure

grayscale values to embed data into the image. However, side information for zero point and peak point should be transmitted to the receiving side for embedded data retrieval. Lee *et al.* [7] exploited the histogram of difference image between odd and even lines to embed more data. In addition, there is no need to transmit any side information to the receiving side for data retrieval.

Although these reversible data hiding algorithms using histogram modification make enough space for data hiding and generate good visual quality after data embedding, these algorithms don't consider problems of underflow and overflow. To overcome the problems of underflow and overflow, we propose a new reversible data hiding algorithm based on histogram modification of block image. Section 2 and 3 describe the embedding and extracting algorithm for the proposed reversible data hiding algorithm, respectively. In section 4, simulation results are compared with other reversible data hiding algorithms. Finally, conclusions are drawn in section 5.

2 Proposed Data Embedding Algorithm

To protect the underflow and overflow problems, we proposed a new reversible data hiding scheme based on histogram modification of non-overlapped block images. Because of systemic insertion of 128 bits hash codes in partition images, our method has the merit of verification of integrity in addition to the reversibility. Data embedding procedure is consists of four main steps: partitioning & making watermark, classifying block, and shifting & Embedding.

2.1 Partitioning and Making Watermark

Given an 512×512 image, partitioning of the image into non-overlapping blocks constitutes the lowest level of the hierarchy as shown in Fig. 1. To make space for embedding the 128 bits hash code and payload data, four level hierarchical block structure is constructed.

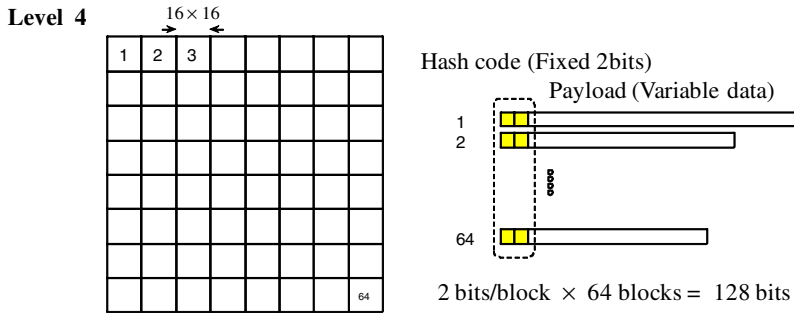


Fig. 2. Embedded Data Construction in the Block

The 128 bits hash code will be used for the verification of integrity. The 128 bits hash code is generated by MD5 algorithm with the original input 512×512 image. MD5 algorithm generates a unique fixed length output of 128 bits from a variable length message.

Fixed 2 bits hash code which are regularly separated from the 128 bits hash code and variable payload data will be inserted into every 16×16 block as shown in Fig. 2. The size of payload data is variable because embedding space is variable according to the size of peak histogram of block image.

The same hash code is inserted into every 128×128 block with raster scan order. Consequently, the same hash code is embedded sixteen times repeatedly. Due to this systematic structure, we can find out whether embedded image have a distortion or not after extraction of embedded data.

2.2 Classifying Block

To embed the watermark data in the 16×16 block, enough space should be reserved at the both end sides of the histogram. To protect the underflow and overflow problems, we need 6 empty points at the both end sides at least.

According to the shape of histogram, the block is categorized into three classes as shown in Fig. 3.

- Class 1: The shifting and embedding is done.
- Class 2: Only shift operation is done. Class 2 should be required to distinguish whether the watermark is embedded or not when some hist exits at the both end sides.
- Class 3: Don't execute any operation but keep intact to avoid underflow and overflow problems.

2.3 Shifting and Embedding

In the histogram of block image, we first find a peak point. A peak point corresponds to the grayscale value having the maximum number of pixels in the given block image. To avoid the side information transmission problem of [5][6], we use two points on both sides of peak point.

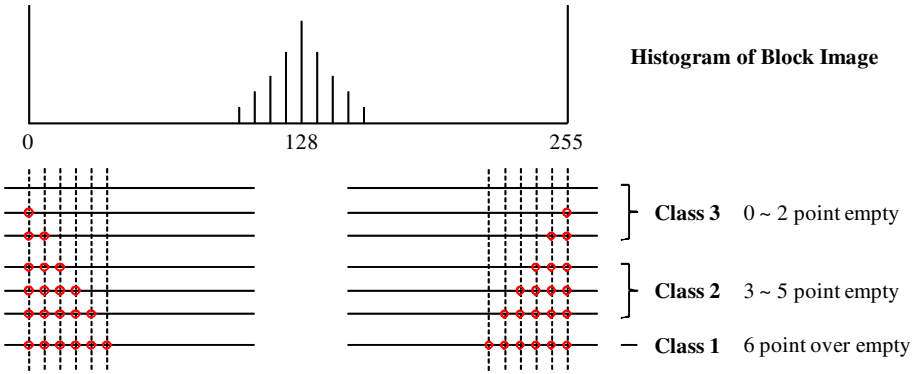


Fig. 3. Histogram of Block Image and Classifying Block

Next, the block image is scanned in a raster scan order. During the scanning process, the grayscale value of pixels between “peak point + 2” and 249 is incremented by “3”. In addition, the grayscale value of pixels between 6 and “peak point - 2” is decremented by “3” as shown in Fig. 4(b). This step is equivalent to shifting the range of the histogram to the right-hand side and left-hand side by 3 unit, respectively.

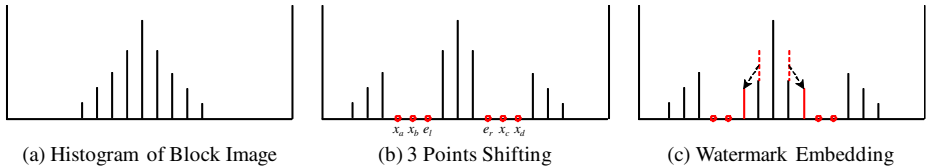


Fig. 4. Histogram Shift and Embedding

To embed the watermark data, the block image is scanned once again in the same sequential order. Once a pixel with grayscale value of “peak point + 1” is encountered, we check the first bit of watermark data. If the corresponding bit is “1”, x_c is incremented by 1. Otherwise, x_d is incremented by 1. To compensate increment at x_c or x_d , the value of “peak point + 1” is decremented by 1. From the second bit of watermark data, if the bit to be embedded is “1”, the e_r is incremented by 1. Otherwise, the pixel value remains intact.

If the value of the “peak point + 1” is entirely exhausted or originally zero, we use the left side of the peak point. To use the grayscale value of “peak point - 1”, the block image is scanned once again in the same sequential order. The left side embedding procedure is similar to that of the right side.

The embedding rules are summarized in Table 1. The meaning of abbreviation is as follows: RSE (right side embedding), LSE (left side embedding), FEDO (first embedded data is one), FEDZ (first embedded data is zero).

Table 1. Embedding Rules

$x_a x_b x_c x_d$	Meaning
0 0 0 1	RSE & FEDO
0 0 1 0	RSE & FEDZ
0 1 0 0	LSE & FEDZ
1 0 0 0	LSE & FEDO
1 0 0 1	RSE & FEDO , LSE & FEDO
0 1 1 0	RSE & FEDZ , LSE & FEDZ
1 0 1 0	RSE & FEDZ , LSE & FEDO
0 1 0 1	RSE & FEDO , LSE & FEDZ

When we complete embedding procedure, we get the modified histogram as shown in Fig. 4(c). During explanation of embedding procedure, we can know why the payload data is variable. The size of the embedded data is dependent on the size of the “peak point ± 1 ”. If we meet the class 2 or class 3 during embedding process, insertion of the corresponding 2 bits of hash code is skipped.

3 Proposed Data Extraction Algorithm

3.1 Classifying Block of Watermarked Block Image

It is not easy to distinguish classes because the block histograms resemble each other in shape after shifting and embedding data. However, it is possible to classify the classes by analysis of the shape around the “peak point” and both end sides. The method for classification is summarized in Table 2. For example, if the block histogram has the 0 2 empty points at the both end sides and $x_a, x_b, e_l, e_r, x_c, x_d$ equal to zero, this block is class 2 because only shift operation is done in class 2.

3.2 Data Extraction and Recovery

Extraction is only applicable to the Class 1. To understand the extraction algorithm more clearly, it is presented in terms of pseudocode.

1. Right side check

- (a) Scan the block image in the same sequential order as that used in the embedding procedure.

Table 2. Classifying Rules

	Both End Sides	Around Peak Point
Class 1	3 over empty points	$x_a, x_b, e_l, e_r, x_c, x_d \neq 0$
Class 2	0 ~ 2 empty points	$x_a, x_b, e_l, e_r, x_c, x_d = 0$
Class 3	0 ~ 2 empty points	$x_a, x_b, e_l, e_r, x_c, x_d \neq 0$

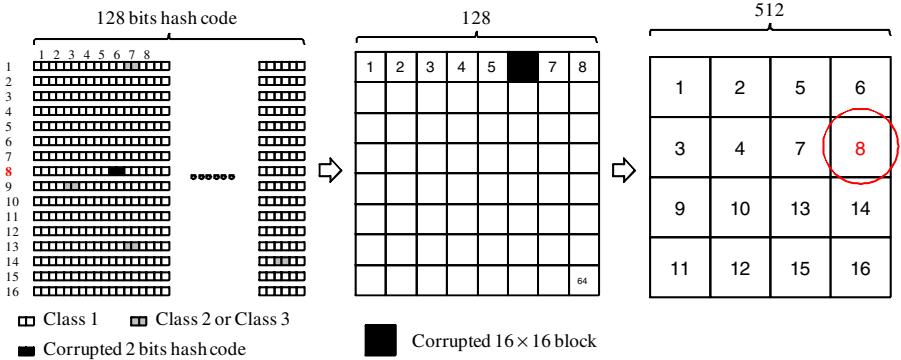


Fig. 5. Verification of Integrity, 6th 16 × 16 block error in the 8th 128 × 128 block

- (b) Check the x_c and x_d around the peak point.
 - i. If $x_c x_d$ is 10, the first extracted bit is “0” and subtract 2 from the current pixel value for recovery.
 - ii. If $x_c x_d$ is 01, the first extracted bit is “1” and subtract 3 from the current pixel value for recovery.
 - iii. After parsing $x_c x_d$
 - A. whenever we meet “peak point + 1” pixel value, we extract “0”
 - B. whenever we meet “ e_r ” pixel value, we extract “1” and subtract 1 from the current pixel value for recovery.
- (c) If $x_c x_d$ is 00, go to “Left side check”

2. Left side check

- (a) The block image is scanned once again in the same sequential order.
- (b) Check the x_a and x_b around the peak point.
 - i. If $x_a x_b$ is 10, the first extracted bit is “1” and add 3 to the current pixel value for recovery.
 - ii. If $x_a x_b$ is 01, the first extracted bit is “0” and add 2 to the current pixel value for recovery.
 - iii. After parsing $x_a x_b$
 - A. whenever we meet “peak point - 1” pixel value, we extract “0”
 - B. whenever we meet “ e_l ” pixel value, we extract “1” and add 1 from the current pixel value for recovery.

3. Assemble the extracted data

- (a) Gather sixteen 128 bits hash codes
- (b) Concatenate payload data

3.3 Verification of Integrity

After gathering sixteen hash codes, we can verify the integrity of recovered image as shown in Fig. 5. Because the same hash code is repeatedly inserted in the every 128×128 block, if some bits of the concatenated hash code are different from other hash codes, the corresponding 16×16 block can be corrupted by attack. In class 2 and class 3, we cannot decide because the the fixed 2 bits hash code was not embedded.

4 Experimental Results

In order to evaluate the performance of the proposed scheme, we perform computer simulations on many 8-bits grayscale images with 512×512 pixels. Fig. 6 shows the original and embedded images. It is observed that there is no visible degradation due to embedding in the watermarked images.

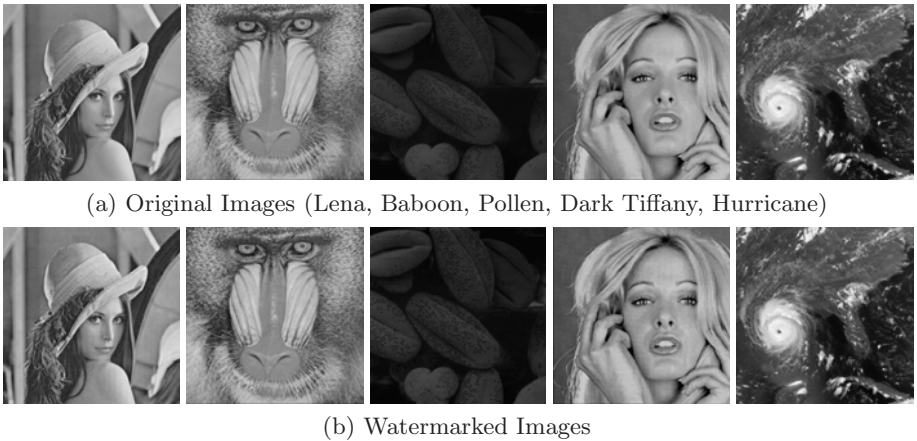


Fig. 6. Original and Watermarked Images

Table 3. Simulation Results for the Proposed Algorithm

Images	Class 1	Class 2	Class 3	Capacity	PSNR
Lena	1024	0	0	2048+27714	39.49
Baboon	1024	0	0	2048+15689	39.13
Pollen	840	161	23	1680+31450	40.70
Tiffany	897	4	123	1794+17214	39.96
Hurricane	864	15	145	1728+20916	40.10

Table 3 summarizes the experimental results of the proposed algorithm. Our proposed reversible data hiding algorithm is able to embed about $17.7 \sim 33.1$ kbits into a $512 \times 512 \times 8$ grayscale image while guaranteeing visual quality. Performance comparison with other reversible data hiding algorithms in terms of

Table 4. Performance Comparison with Other Algorithms

Images	Ni [6]		Lee [7]		Proposed	
	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
Lena	2723	53.62	23692	52.55	29762	39.49
Baboon	4322	51.34	18533	52.29	17737	39.13
Pollen	16580	48.28	28795	53.94	33130	40.70
Tiffany	4301	<i>24.81</i>	16465	<i>30.95</i>	19008	39.96
Hurricane	2928	47.40	17816	<i>27.66</i>	22644	40.10

capacity and PSNR is presented in Table 4. The results of Ni *et al.* [6] is obtained by one zero point and one maximum point. Italic type means that underflow and overflow problems have occurred in watermarked image. In case of underflow and overflow problems, the value of PSNR is considerably decreased.

5 Conclusions

We have proposed a reversible data hiding algorithm based on the histogram modification of block image. To solve the underflow and overflow problems, we embed the watermark data into a selected block image. Experimental results showed that the proposed scheme provides high embedding capacity while keeping low distortion in watermarked image and overcoming the overflow and underflow problems. In addition, we can verify the integrity by comparing the extracted hash codes. It is expected that the proposed reversible data hiding algorithm having these properties can be deployed for a wide range of applications which requires the original image.

Acknowledgment

This work was supported by the IT R&D program of MKE/MCST/IITA. [2009-S-017-01, Development of user-centric contents protection and distribution technology].

References

1. Fridrich, J., Goljan, M., Du, R.: Invertible Authentication. In: Proc. SPIE Security and Watermarking of multimedia Contents III, vol. 4314, pp. 197–208 (2001)
2. Tian, J.: Reversible Data Embedding Using a Difference Expansion. IEEE transactions on circuits and systems for video technology 13(8), 890–896 (2003)
3. Alattar, A.M.: Reversible Watermark Using the Difference Expansion of Triplets. In: Proc. Int. Conf. Image Processing, vol. 1, pp. 501–504 (2003)
4. Alattar, A.M.: Reversible Watermarking Using the Difference Expansion of a Generalized Integer Transform. IEEE Transactions on Image Processing 13(8), 1147–1156 (2004)

5. Ni, Z., Shi, Y.Q., Ansari, N., Su, W.: Reversible Data Hiding. In: Proc. Int. Symp. Circuits Sys., pp. 912–915 (2003)
6. Ni, Z., Shi, Y.Q., Ansari, N., Su, W.: Reversible Data Hiding. IEEE Transactions on circuits and systems for video technology 16(3), 354–362 (2006)
7. Lee, S.K., Suh, Y.H., Ho, Y.S.: Lossless Data Hiding Based on Histogram Modification of Difference Images. In: Aizawa, K., Nakamura, Y., Satoh, S. (eds.) PCM 2004. LNCS, vol. 3333, pp. 340–347. Springer, Heidelberg (2004)

A Rock Structure Recognition System Using FMI Images

Xu-Cheng Yin¹, Qian Liu¹, Hong-Wei Hao¹, Zhi-Bin Wang¹, and Kaizhu Huang²

¹ Department of Computer Science, School of Information Engineering,
University of Science and Technology Beijing, Beijing 100083, China

² Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China
xuchengyin@ies.ustb.edu.cn, lq60001q@163.com,
hhw@ies.ustb.edu.cn, wzb1818@yahoo.cn, kzhuang@nlpr.ia.ac.cn

Abstract. Formation Micro Imager (FMI) can directly reflect changes of wall stratum and rock structures. It is also an important method to divide stratum and identify lithology. However, people usually deal with FMI images manually, which is extremely inefficient and may incur heavy burdens in practice. In this paper, with characteristics of rock structures from FMI images, we develop an efficient and intelligent rock structure recognition system by engaging image processing and pattern recognition technologies. First, we choose the most effective color and shape features for rock images. Then, the corresponding single classifier is designed to recognize the FMI images. Finally, all these classifiers are combined to construct the recognition system. Experimental results show that our system is able to achieve promising performance and significantly reduce the complexity and difficulty of the rock structure recognition task.

Keywords: FMI, rock structure, feature extraction, multiple classifier system.

1 Introduction

As the oil and gas exploration becomes gradually complicated, the traditional well logging method has many problems such as they are difficult in recognizing effective layers and also hard to estimate reserves parameters. These problems seriously influence the objectivity of reserves assessments. In contrast, Formation Micro Imager (FMI) technology can provide rich information on fractured reservoirs, and most importantly it can be applied to identify fractured reservoirs qualitatively and can help explain them quantitatively [1-2]. Some FMI image samples are as shown in Fig.1.

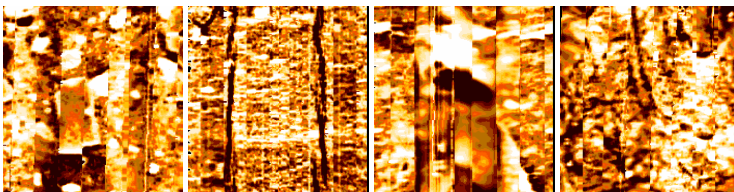


Fig. 1. FMI images

However, the most commonly way to deal with FMI images in China is still dependent on manual processing, which hence incurs heavy workload and inefficiency. Moreover, the results obtained manually are usually affected by the experience of the operators and may not be consistent in practice. Furthermore, it is fairly difficult to make fully use of FMI images for agencies without enough expertise and experienced geologists [3]. Hence, it is of great significance to develop an efficient intelligent recognition system using FMI images, for promoting oil and gas exploration.

In this paper, by applying pattern recognition technologies, we develop a rock structure recognition system based on FMI images. This system extracts useful features effectively and then recognizes rock structures with FMI images automatically. First, we chose the color and shape features for rock images. Then, the corresponding single classifier is designed to recognize the FMI images. Finally, all these classifiers are combined to construct the recognition system.

The rest of the paper is organized as follows. Section 2 describes the recognition system framework. And experimental results of the recognition system are shown in Section 3. Finally, some conclusions are drawn in Section 4.

2 System Framework and Analysis

2.1 System Framework

Rock structures can be classified by the proposed recognition system with image processing and pattern recognition technologies. This system includes three main modules, i.e., image pre-processing, feature extraction, and structure recognition, which is shown in Fig.2.

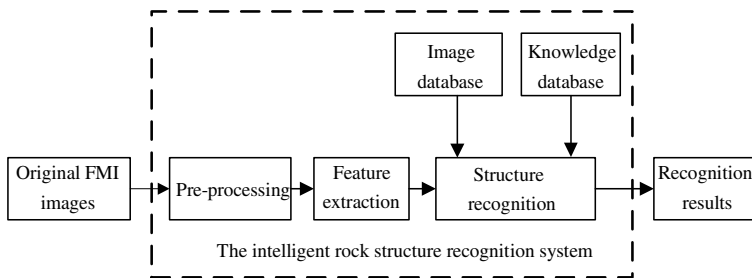


Fig. 2. The rock structure recognition system framework

The image pre-processing step is to filter out the noise. The main task for feature extraction is to extract characteristics of the rock, such as color and shape features. Structure recognition applies and combines multiple classifiers (k-nearest neighbor classifiers). Image database and knowledge database mainly provide the references for recognizing images and categories of the rock structures. In this paper, we will mainly focus on feature extraction and the combination of multiple classifiers.

2.2 Feature Extraction

Feature extraction is a very important step, and it heavily affects the final recognition accuracy. Following many other systems, we select color and shape features for our recognition system.

In fact, shape describes the important difference among the rocks with different structures. Traditional methods merely exploiting color features cannot efficiently categorize rocks. As a result, combing color and shape features can largely improve the recognition accuracy. In this paper, rock structure features include the proportion of white color accounted in the image and the shape information.

2.2.1 Color Feature Extraction

From different rock structures, we find that particles distribute uniformly and are always white, which is an important feature of some rock structures.

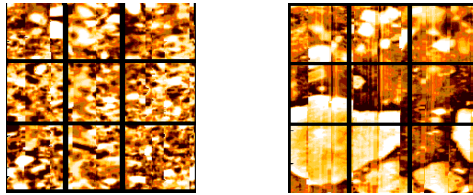


Fig. 3. Particle distribution comparison

As it is shown in Fig.3, the particles in the left image are uniform in distribution, while the particles in the right one are very uneven in distribution. Consequently, we propose the following color feature extraction method. The block FMI image is divided into blocks and the white color proportion in every one is counted. After that the proportion values between the blocks are compared and calculated as features. If the values are even, the distribution is uniform; otherwise, it is not uniform.

2.2.2 Shape Feature Extraction

To a large extent, shape features can reflect the structure information of objects, and most efficient features for classifying are mainly composed of shape characteristics [4-5]. There are a lot of methods for shape feature extraction. In the early experiments we tried to use traditional edge detection methods to extract structure information. The FMI images are processed with Sobel and Canny transformation, and then the edges of the images are acquired. The results are as shown in Fig.4. From Fig.4, some broken edges can be detected. However, some false edges are also detected. Considering the nature of FMI images and also motivated from idea exchange with geologists, we propose a more simple and efficient way to extract shape features of rock, which includes puncture, tour and projection steps.

Puncture is a method proposed for rhyolite and crack rock. The image is scanned progressively either by row or by column, and a puncture occurs when scanning through a rhyolite or crack structure. This feature is the times that puncture happens in the whole image.

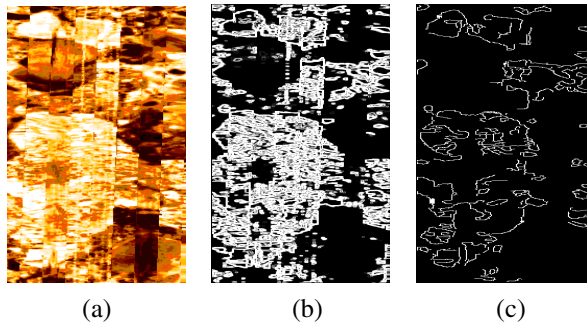


Fig. 4. Fig(a) is the original FMI image, Fig(b) is the image processed by Sobel transformation, and Fig(c) is the image processed by Canny transformation

Tour means to scan the image progressively by row or by column. When the line enters a block, a tour starts to count the pixels until it gets out of the block. This feature is the average count of the tour, which means the total count divided by the number of blocks.

Projection is to divide the image into several blocks in proportion, project all the values of the color to the bottom, and accumulate them into one value in every block. This feature is very useful for the rock which texture is very uniform or very loose in distribution.

The features mentioned above can be calculated with different segmentation ways (by row, by column, or by block). And these shape features can be also used with color features. Consequently, all these features can be grouped into different feature vectors for rock classification.

2.3 Classifier Design and Integration

There are several commonly used classifiers: the minimum distance classifier, the nearest neighbor classifier, the k-nearest neighbor classifier, and the BP neural network classifier.

2.3.1 The Minimum Distance Classifier

The minimum distance classifier uses a base template in the feature space to represent a pattern, and the classification is based on the distance between the feature vector of the sample to be identified and this template.

If M_i is the base template of pattern class ω_i ($i = 1, 2, \dots, C$):

$$M_i = (m_{i_1}, m_{i_2}, \dots, m_{i_n})^T, i = 1, 2, \dots, C. \quad (1)$$

And if X is the feature vector of the sample to be identified:

$$X = (x_1, x_2, \dots, x_n)^T, \quad (2)$$

Here, $d(X, M_i)$ is the distance between the sample X and M_i , the base template of the pattern class ω_i , and principle is that if $d(X, M_i)$ is the smallest value in all distances, the sample belongs to pattern class ω_i . Practices proved that this is a very simple and effective method.

2.3.2 The Nearest Neighbor Classifier and the K-Nearest Neighbor Classifier

Assume that there are C pattern classes $\omega_1, \omega_2, \dots, \omega_c$, and there are N_i samples in each corresponding class, whose pattern classes are known, where $i = 1, 2, \dots, C$.

The nearest neighbor classifier uses all the samples in every pattern class as representative points, and classifies the unidentified sample X into the class whose samples are nearest to it. Therefore, the nearest neighbor classifier can partially resolve the influence caused by the differences among the sample even vectors. The discrimination function for pattern class ω_i is:

$$g_i(X) = \min_k \|X - X_i^k\|, k = 1, 2, \dots, N_i \quad (3)$$

where the i in X_i^k means pattern class ω_i , and k means the sample k in ω_i . If Eq. (4) is satisfied, then $X \in \omega_j$.

$$g_j(X) = \min_i g_i(X), i = 1, 2, \dots, c \quad (4)$$

The k-nearest neighbor classifier is a general version of the nearest neighbor classifier. The principle is to find k nearest samples the closest to unidentified X , and it belongs to the pattern class that most of the k samples belong to. Taking into account the efficiency decrease caused by classifiers integration and that the features selected in every layer can discriminate the rock in some extent, hence this system adopts simple and practical classifiers: the nearest neighbor classifier and the k-nearest neighbor classifier.

2.3.3 Classifier Combination

Single classifiers can be integrated into a final one. Generally, there are three methods to integrate them according to the decision-making information provided by each classifier.

The first one is decision output-oriented integration method. Although the information outputted by classifiers is very little, it is still commonly used and other forms of output can be transformed to this one. The second one is sorting output oriented integration method. This type of approaches first sorts the categories by its possibility according to the output, and then integrates them based on various strategies. The third one is the measure output-oriented method. This method exports a measure value for every category, such as probability, confidence level, or distance measure [6].

With FMI images, this rock structure recognition system is supposed to classify the rock into five types of structures: lava, tuff, tuff breccia, volcanic breccia, and ablation breccia. All these types of structures are very complex, and a single feature is not

sufficient for such a complex task. Accordingly, this system needs to recognize rock structure by many effective features [7].

In this paper, for every type of rock structure, a specific feature vector is selected, and the classifiers are integrated hierarchically. Every layer uses a specific feature vector to recognize rocks of specific structure. If the rock is not recognized at the current layer, it will go on to the next layer until it is finally identified [8]. In order to get better performance, a voting mechanism is adopted based on the k-nearest neighbor classifier [9-10].

2.4 System Workflow

The system workflow is as shown in Fig.5, and the process is described as follows:

Step 1: Pre-process the image to be classified.

Step 2: Extract the features from the processed image.

Step 3: Recognize the image. If it is lava, go to Step 8. Else, go to Step 4.

Step 4: Recognize the image. If it is tuff, go to Step 8. Else, go to Step 5.

Step 5: Recognize the image. If it is tuff breccia, go to Step 8. Else, go to Step 6.

Step 6: Recognize the image. If it is volcanic breccia, go to Step 8. Else, go to Step 7.

Step 7: Recognize the image. If it is ablation breccia, go to Step 8. Else, classify it into some appointed type directly then go to Step 8.

Step 8: Export the result of recognition.

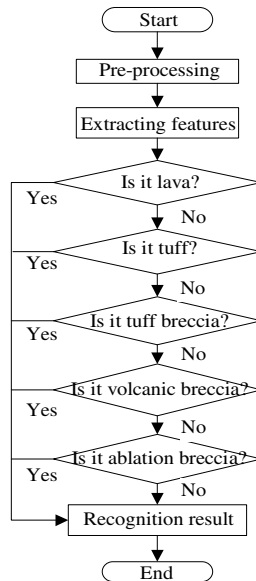


Fig. 5. System workflow

3 Experiments

In these experiments, we classify rock structures with some real FMI rock images obtained by the oil and gas exploration. Because of high cost of image capture and secrecy protection, the size of our experiment data is rather small (shown in Table 2). The experiments include two parts: one is for feature selection and the other is for rock classification. The results of feature selection are shown in Table 1, where we can find that different optimal feature vectors for different situations.

Table 1. The optimal feature vector

Rock Structure	Feature	Zoning Mode	Quantity of Zones
Lava	Black Vertical Puncture	Vertically	60
Tuff	Black Horizontal Puncture	Horizontally	5
Tuff Breccia	White Pixels Rate	By block	5*2
Volcanic Breccia	White Vertical Tour	Vertically	5
Ablation Breccia	Projection	Vertically	25

Then we use the above optimal features as a feature vector for rock classification. The recognition results can be seen in Table 2.

Table 2. The experimental results of this system

Rock Structure	Number of Training Set	Number of Testing Set	Classified correct	Classified wrong	Accuracy Rate %
Lava	1	34	29	5	85.3
Tuff	1	16	12	4	75.0
Tuff Breccia	1	12	10	2	83.3
Volcanic Breccia	1	16	13	3	81.3
Ablation Breccia	1	12	9	3	75.0

It can be concluded from Table 2 that with this optimal feature vector, the average accuracy rate of this system is above 80%. We can also see that, the vector partially reflects the structure characteristics. For example, the vertical puncture is corresponding to the rhyolite structure of lava. This feature is important and proves critical for recognizing lava. The accuracy rate is satisfying and can meet the demands of geologists in the oil and gas exploration. Note that, we do not compare our system with other competitive algorithms because we rarely see any intelligent rock structure recognition systems in the literatures.

In our experiments, we only use one training sample for each category. Obviously, the recognition performance can be largely improved with more training samples. As shown in Table 2, the classification accuracy of Tuff and Ablation Breccia rocks are only 75%. And another possible improvement is to investigate more effective features so as to achieve higher accuracy.

4 Conclusions

In this paper, we utilized the characteristics of rock structures from FMI images and developed an efficient intellectual rock structure recognition system using image processing and pattern recognition technologies. The recognition system is able to select useful color and shape features, and adopt multiple classifiers for the final decision of a rock structure. Experiments with real FMI images captured from the oil and gas exploration showed that our system can largely reduce the complexity and difficulty of the recognition of rock structure, and effectively raise the automatic level of exploration. Some further issues include collecting more training samples and exploiting more effective features.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No.60675006 and thanks Fei Wu for his contributions to this paper.

References

1. Singh, U., Van Der Baan, D.: FMS/FMI borehole imaging of carbonate gas reservoirs, Central Luconia Province, offshore Sarawak, Malaysia. In: 1994 American Association of Petroleum Geologists (AAPG) international conference and exhibition, Kuala Lumpur, Malaysia, pp. 1162–1163. AAPG Bulletin (2001)
2. Laubach, S.E., Gale, J.F.W.: Obtaining Fracture Information for Low-Permeability (Tight) Gas Sandstones from Sidewall Cores. *Journal of Petroleum Geology* 29(2), 147–158 (2006)
3. Endres, H., Lohr, T., Trappe, H.: Quantitative fracture prediction from seismic data. *Petroleum Geoscience* 14, 369–377 (2008)
4. Payenberg, T.H.D., Lang, S.C., Koch, R.: A Simple Method for Orienting Conventional Core Using Microresistivity (FMS) Images and a Mechanical Goniometer to Measure Directional Structures on Cores. *Journal of Sedimentary Research* 70, 419–422 (2000)
5. Russell, S.D., Akbar, M., Vissapragada, B., Walkden, G.M.: Rock Types and Permeability Prediction from Dipmeter and Image Logs: Shuaiba Reservoir (Aptian), Abu Dhabi. AAPG Bulletin 86, 1709–1732 (2002)
6. Roli, F., Giacinto, G., Vernazza, G.: Methods for Designing Multiple Classifiers Systems. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 78–87. Springer, Heidelberg (2001)
7. Chen Lei, K.: A generalized adaptive ensemble generation and aggregation approach for multiple classifier systems. *Pattern Recognition* 42(5), 629–644 (2009)
8. Kang, H.J., David, D.: Selection of classifiers for the construction of multiple classifier systems. In: 8th International Conference on Document Analysis and Recognition, Seoul, Korea, pp. 1194–1198 (2005)
9. Ruta, D., Gabrys, B.: Classifier selection for majority voting. *Information Fusion* 6(1), 63–81 (2005)
10. Ruta, D., Gabrys, B.: Application of the evolutionary algorithms for classifier selection in multiple classifier systems with majority voting. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 399–408. Springer, Heidelberg (2001)

Analyzing Price Data to Determine Positive and Negative Product Associations

Ayhan Demiriz¹, Ahmet Cihan², and Ufuk Kula³

¹ Dept. of Industrial Engineering
Sakarya University
54187, Sakarya, Turkey
ademiriz@gmail.com

² Dept. of Industrial Engineering
Kocaeli University
Kocaeli, Turkey
ahmet.can.cihan@gmail.com

³ Dept. of Industrial Engineering
Sakarya University
54187, Sakarya, Turkey
ufukkula@gmail.com

Abstract. This paper presents a simple method for mining both positive and negative association rules in databases using singular value decomposition (SVD) and similarity measures. In literature, SVD is used for summarizing matrices. We use transaction-item price matrix to generate so called ratio rules in the literature. Transaction-item price matrix is formed by using the price data of corresponding items from the sales transactions. Ratio rules are generated by running SVD on transaction-item price matrix. We then use similarity measures on a subset of rules found by Pareto analysis to determine positive and negative associations. The proposed method can present the positive and negative associations with their strengths. We obtain subsequent results using cosine and correlation similarity measures.

1 Introduction

Data mining is used for discovering knowledge from large databases. As being an interdisciplinary approach data mining utilizes algorithms developed in computer science, mathematics, artificial intelligence, machine learning, statistics, optimization and other fields. As one of the early tools of recommender systems [1] Apriori algorithm [2] has been widely used for finding positive item associations. Apriori algorithm searches for the relations between product groups satisfying user supplied support and confidence levels and finds frequently bought product groups by customers. Although Apriori algorithm and its variations mostly use transactional data format, some forms of it require the data in transaction-item matrix format. Basically this type of matrix consists of binary data. Transaction-item (user-item) matrix is also the source of the data used in various collaborative filter based recommender systems.

Like the main stream research in association mining, item price data have long been neglected in recommender systems for finding relations between items. From this point of view, Apriori algorithm has a disadvantage of omitting the price paid by the customers to the purchased products despite of readily available data in transactions. This paper explores possibility of using transaction-item price data to find relationships (associations) between items. An early work, [3], studies ratio rules derived from the expense data to understand how much money would be spent on an item compared to the other items. In [3], a sample supermarket expense dataset was used in constructing the discussions for the ratio rules. Singular Value Decomposition (SVD) is used for finding the ratio rules which simply are eigenvectors corresponding to eigenvalues.

Similarly we adopt transaction-item price dataset from apparel retailing to assess the usability of price data for finding item relations. Our ultimate goal is to use these results in determining cross-price elasticities among multiple items. However our early findings indicate that we can use these results for determining both positive and negative item relationships as well. Our approach is summarized as follows: We first use SVD to decompose the transaction-item price matrix to find the eigenvectors i.e. ratio rules. We then deploy Pareto analysis to determine the important rules. This is indeed equivalent to picking the most influential eigenvalues and their eigenvectors. We then utilize some similarity measures, specifically cosine and correlation coefficient, to determine the sign and strength of relationships between items.

We also compare the outcome of our approach with traditional association mining results in this paper. We show that some of the positive associations can be recovered by our approach, however some associations are not found by our approach. This is indeed an important indication of the price sensitivity of the associations. Meaning that if the prices items high at the beginning, which is the case for the apparel retailing, items are more likely purchased alone. However the prices of the items are reduced as season progresses and as the prices of the items are marked down appropriately, it becomes more likely that certain items would be purchased together. This will obviously contribute a positive affect on the associations among such items.

Our aim in this paper is to show that item price data could potentially useful in determining positive and negative relationships between items. We summarize the contributions of the paper in the remaining of this section.

1.1 Contributions

The following contributions are provided in this paper:

- Transaction-item price matrix has been utilized in an association mining framework,
- Positive and negative relationships can be found by using transaction-item price matrix,
- Evidence is presented that positive associations can be attributed to the price reductions.

As listed above the paper has three main contributions. The rest of the paper is structured as follows. In Section 2, we give a brief description of the preliminaries. In Section 3, we introduce our methodology. A short illustrative example is presented in Section 4. We present results of our approach on a real dataset coming from apparel retailing in Section 5. We then conclude our paper in Section 6.

2 Preliminaries

Ratio rule mining technique uses eigensystem analysis. We can use SVD to find eigenvalues and eigenvectors of a non-square matrix. The number of eigenvalues of a matrix is equal to rank of this matrix. SVD method can simply be described for the matrix X , with transaction (customer) information in rows and product information in columns, by the following formula:

$$X = U \times \Lambda \times T' \quad (1)$$

U and T are orthonormal matrices called left and right singular values respectively. Λ is the diagonal matrix with eigenvalues of X corresponding amplitude of eigenvectors described by T . All of the eigenvectors described by T are not used as ratio rules. There is a heuristic method for determining which eigenvalues are accepted for ratio rules [3]. According to this heuristic method the cutoff for the rules is %85 of the cumulative sum of eigenvalues. If the leading eigenvectors are very significant then using the rest of them as rules is unnecessary. Thus we can find a cutoff level for the rules by using Pareto analysis.

Pareto analysis is fundamentally using Pareto principle which can simply be phrased as follows: %80 of produced outputs are from %20 of inputs. To find which inputs have strong effects to generate the outputs you can plot the graph of inputs to corresponding outputs. This paper utilizes Pareto analysis as the number of eigenvectors in inputs and eigenvalues as outputs. The worst case scenario is that the eigenvalues are all equal. In this case, the Pareto plot has a slope of 45 degrees. For this reason, the cutoff level is determined by the slope of the line segments where the slope is lower than of 45 degrees. In other words if a line segment has a slope lower than 45 degrees it can be considered as the cutoff point for the rules.

3 The Methodology

Generating eigenvectors i.e. ratio rules is a straight forward step in our framework. After determining the most significant rules (i.e. truncated SVD) by Pareto analysis, we can deploy some similarity measures to summarize the relationships between products. In the literature there are many similarity measures [4] used for many different problem types.

The purpose of this paper is to find both positive and negative relationships (similarities) between products on significant rules found by SVD. There are two types of important information embedded in these rules. The first one summarizes the amount (i.e. ratio) of price paid, which represents the general behavior.

The second one is the sign information which represents the direction of the relationships between products. Therefore, we can deploy transaction-item price data to find the relationships between products.

One could use more measures to find similarities, however, for the brevity of the study we use two of them: correlation and cosine. Correlation and cosine similarity measures can vary between -1 and 1. If the value of the measure is negative, this means that the products have negative association between them. If the value of the measure is near zero then it can be concluded that the products are not related. Otherwise, if the value of the similarity measure is positive, then it can be concluded that the products have positive association between them. We give brief definitions of these similarity measures below.

3.1 Correlation Similarity Measure

Correlation coefficient similarity measure can be expressed by the following equality: $\rho(x, y) = \frac{\sigma_{xy}^2}{\sigma_x \sigma_y}$ where σ_{xy}^2 represents covariance between vector x and vector y and σ_x represents the standard deviation of vector x . Correlation coefficient is a widely used statistic in determining significant linear relationships.

3.2 Cosine Similarity Measure

Cosine similarity measure depends on the degree between two vectors. Cosine similarity measure can be expressed by equality: $\cos(x, y) = \frac{x \cdot y}{\|x\| \|y\|}$ where $x \cdot y$ is the dot product between vector x and vector y . $\|x\|$ is the 2-norm of vector x .

4 An Illustrative Example

In the following, we present an illustrative example (see Table 1) to depict our methodology. Each transaction is composed of price paid for three products. Notice that this is just a sample data. Obviously, if a product is not purchased at all then the corresponding price is equivalent to 0. We can potentially consider this dataset as an expense dataset, since the numbers correspond to the amount paid. However in our study we prefer to call it transaction-item price matrix. Table 1 lists the sample data used in this section.

After applying SVD to the data matrix, we find the eigenvalues and eigenvectors as follows:

$$\Lambda = \begin{pmatrix} 11,6123 & 0 & 0 \\ 0 & 7,2180 & 0 \\ 0 & 0 & 3,1708 \end{pmatrix}$$

$$T' = \begin{pmatrix} -0,5036 & -0,6440 & -0,5759 \\ -0,6414 & -0,1678 & 0,7486 \\ 0,5788 & -0,7464 & 0,3286 \end{pmatrix}$$

Table 1. Illustrative Example Data

Transaction	Product 1	Product 2	Product 3
Transaction 1	3	1	0
Transaction 2	2	2	0
Transaction 3	2	1	0
Transaction 4	5	5	0
Transaction 5	0	1	4
Transaction 6	0	2	2
Transaction 7	0	1	2
Transaction 8	0	2	5
Transaction 9	0	3	1
Transaction 10	1	3	4
Transaction 11	4	2	3

Matrix T' corresponds to eigenvectors of matrix X and diagonal of matrix Λ corresponds eigenvalues of corresponding eigenvectors. In Figure 1, the cumulative importance of the rules derived from the eigenvalues is depicted against the number of rules considered. The slopes of the plot indicate the importance of the rules.

Eigenvalue that results in a line segment with a slope under an angle of 45 degrees is the cutoff for rules. However, in order to have a similarity measure we need at least two eigenvectors in our analysis. Since we have three eigenvectors (T'), we can only use two of them for a similarity measure. Notice that if we use all the eigenvectors (ratio rules) in our analysis then the similarity measures, for example cosine, will yield meaningless result that all the products are unrelated. This is due to the fact that all the eigenvectors are orthogonal to each other.

The sample case has the line segment slopes of [1.58 0.98 0.43] corresponding to Rule 1, Rule 2, and Rule 3, respectively. The first line segment has a slope bigger than 1. Technically we should avoid including Rule 2 to our analysis since it has a slope lower than 1. However we need at least two rules to generate similarity measures. The first two rules are given again below.

- Rule 1: [-0.5036, -0.6440, -0.5759]
- Rule 2: [-0,6414, -0.1678, 0.7486]

Based on the above rules we will have the following matrix which can also be called as ratio rules matrix (RR) to determine similarities:

$$RR = \begin{pmatrix} -0.5036 & -0.6440 & -0.5759 \\ -0,6414 & -0.1678 & 0.7486 \end{pmatrix}$$

The columns of the ratio rules matrix above correspond to the products. Similarity measures can be calculated by using this matrix to determine product relations (similarities). Using similarity measures over columns of ratio rules (rule-product) matrix results in product to product similarities. Notice that we

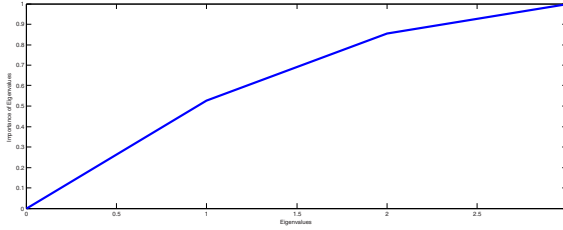


Fig. 1. Pareto Graph for Toy Example

can also use the sign of each value of the ratio rules matrix for determining product similarities. In this case we will have the following discretized ratio rules matrix to determine product similarities.

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & -1 & 1 \end{pmatrix}$$

After applying cosine based similarity measure on ratio rules matrix *RR* above, we get the following product similarities for the sample problem:

$$\begin{pmatrix} 1 & 0,7959 & -0,2469 \\ 0,7959 & 1 & 0,3901 \\ -0,2469 & 0,3901 & 1 \end{pmatrix}$$

A correlation coefficient measure will be as follows:

$$\begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{pmatrix}$$

If we use a discretized ratio rules matrix on cosine based similarity measure, this will yield the following product similarities:

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Similarity measures over continuous data shows that product 1 and product 3 have negative association between them. Applying the correlation coefficient similarity measure on discrete rules yields inconclusive results. Similarity measures over discrete rules are inconsistent, because there is no variation (univariate) in other words the standard deviation is equal to 0.

5 Analysis

We use the sales data of summer season of year 2007 from a leading apparel retail firm in Turkey for the analysis. Like in any other retail environment, the

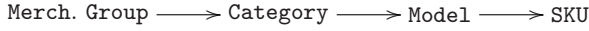


Fig. 2. A Typical Representation of the Product Hierarchy (shown horizontally for brevity)

products in an apparel retail firm can be represented in a hierarchy. A typical hierarchy is shown in Figure 2. The major layers are shown in this hierarchy where merchandise group is shown at the top of the hierarchy, however additional layers can be inserted depending on the structure of the apparel business. Stock keeping unit (SKU) layer is the lowest level in this hierarchy. However SKU level data include unnecessary detail for the analysis. So we decided to use the model level data i.e. the data is aggregated at the size and the color levels for a particular garment.

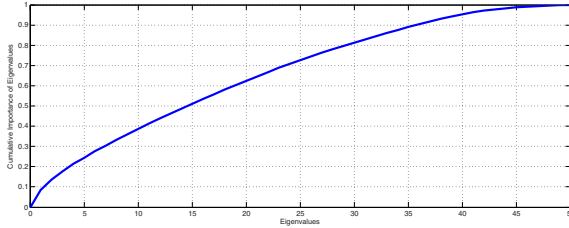


Fig. 3. Pareto Graph of Eigenvalues

For the purpose of this study, we pick the top 50 models out of 710 models belonging to one merchandise group based on the sales figures. Since the firm has provided us data belonging to a particular merchandise group (e.g. women’s apparel), the top 50 models are from the same merchandise group. We then select the transactions with at least 5 products (items) involved (purchased) to reduce the adverse effect of sparsity of the data matrix. This gives us 3,525 transactions from the sales data with 50 models i.e. a 3525×50 data matrix.

There are 50 eigenvalues and corresponding eigenvectors found by using SVD. By applying Pareto analysis and visually inspecting the Figure 3, it is acceptable to conclude that approximately the leading 30 eigenvalues are significant for the given data matrix. We can then calculate the similarity measures to determine the product relationships. It should be noted that the similarity measures used in this paper vary between -1 and 1. In such a scale, measure values near zero (in both directions) represent unrelated products. However there is no clear cut threshold to determine the separation. The lower threshold (nearer zero) is, the more relationships will be found from the similarity measure matrix. For example, in Figure 4, we vary the threshold for the negative relationships between -0.1 and -1. In other words, if we have a similarity value between two products lower than the threshold level (since the similarity in the negative side of the spectrum), we can conclude that these two products have a negative

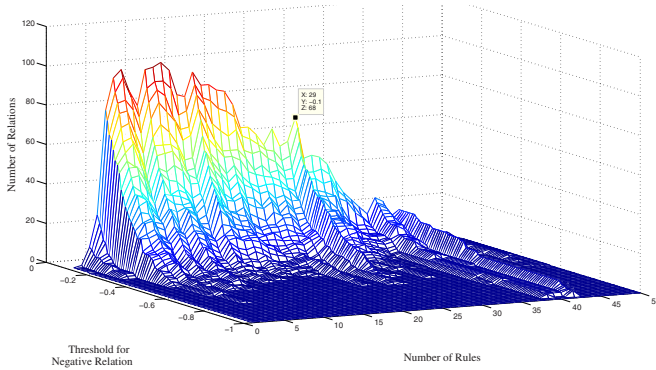


Fig. 4. Number of Negative Relations based on Cosine Similarity

relationship. In the same figure, it is evident that by using the first 29 ratio rules (eigenvectors) with a -0.1 threshold level from cosine similarity measure, we will find 68 negative relationships among 50 products. Correlation coefficient based similarity measure yields 75 negative relationships which has comparable number of relationships with cosine similarity. Out of 68 negative relationships found by the cosine similarity measure, there are only four relationships that could not be accounted by the correlation similarity measure. We can conclude that both measures behave similarly in terms of finding negative relationships. By using the discretized ratio rules, we usually find more relationships than continuous case in our experiments. However these relationships are questionable as seen in the illustrative example given in Section 4.

Similarly, we can vary the similarity threshold to observe the positive relationships as in Figure 5. Recall that a similarity threshold means that any two items which have a positive similarity measure above this threshold are considered similar. For the positive relationships, at 0.1 threshold level cosine similarity measure finds 168 positive relationships by using 29 ratio rules mentioned above. Based on the correlation coefficient similarity measure, our approach finds 177 and 75 positive and negative relationships respectively. Again we use 0.1 and -0.1 threshold levels for the positive and negative relationships respectively. Out of 168 positive relationships found by cosine similarity measure, there are only three relationships that could not be found by the correlation similarity measure which covers 177 positive relationships. These three relations that are not accounted by the correlation similarity measure are the borderline cases i.e. they are just below the threshold level. Again, we can conclude that both cosine and correlation similarity measures behave similarly in terms of finding positive relationships as well.

To compare our approach with the traditional association mining, we apply Apriori algorithm with a support count level 100 which is approximately 2.84% support and 10% confidence levels. We find 73 frequent pairs i.e. positive relationships meaningful. There are 24 pairs overlapping with our approach (cosine

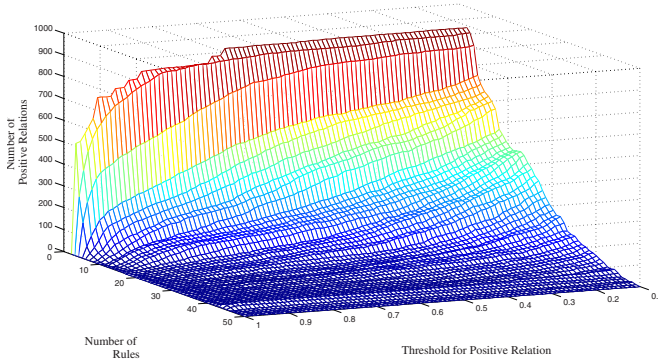


Fig. 5. Number of Positive Relations based on Cosine Similarity

similarity measure at 0.1 threshold level with 168 positive relationships) out of 73 frequent pairs from Apriori algorithm. For the negative association, we utilize indirect association mining from [5] which yields 91 negative relationships. Very few of 68 relationships found by our approach match with the results from indirect association mining.

Discrepancies between the traditional association mining and our approach can be attributed to the price sensitivities (multiple items cross-price elasticities) of the products. In apparel retailing, the price of the items are always higher at the beginning of the season. Later in the season, there might be significant reductions in the prices. When the prices of two items are sufficiently lowered, then the likelihood of purchasing both items increases. If both items are purchased together in a significant level during the sales season, the traditional association mining can pick this behavior as a positive association. However both items can show a different behavior at normal price levels. That's why our approach can identify this relationship as negative, since both items are not usually purchased together at normal prices, but at highly reduced prices.

6 Discussion and Conclusion

We have shown that transaction-item price data can be utilized for finding both positive and negative relationships. We also compare our approach with traditional association mining techniques: Apriori and indirect association mining.

Our analysis indicate that it may not always safe to conclude from a traditional association mining that two items have positive association for all the time, even though they satisfy the minimum support and confidence level constraints. This conclusion might be true if only both items are on sale at significant price reductions. In addition, we should point that the behavior of Apriori algorithm might change drastically at different price levels. To our best knowledge, there are no published results pointing this issue before.

Acknowledgement. This study is supported by the Turkish Scientific Research Council through the grant TUBITAK 107M257.

References

1. Demiriz, A.: Enhancing product recommender systems on sparse binary data. *Journal of Data Mining and Knowledge Discovery* 9(2), 147–170 (2004)
2. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: *SIGMOD Conference*, pp. 207–216 (1993)
3. Korn, F., Labrinidis, A., Kotidis, Y., Faloutsos, C.: Quantifiable data mining using ratio rules. *VLDB J.* 8(3-4), 254–266 (2000)
4. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right interestingness measure for association patterns. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, July 23-26, 2002, pp. 32–41 (2002)
5. Tan, P.N., Kumar, V., Kuno, H.: Using sas for mining indirect associations in data. In: *Western Users of SAS Software Conference* (2001)

Production Planning Algorithm and Software for Sofa Factory

Cholticha Sangngam and Chantana Phongpensri (Chantrapornchai)

Department of Computing, Faculty of Science,
Silpakorn University, Nakorn Pathom, Thailand
ctana@su.ac.th

Abstract. In this paper, we develop an algorithm to create a production plan for a sofa factory. We propose a heuristic to create a schedule. We also model the problem using a linear programming. Both approaches are compared using the real case study in the furniture factory. The heuristic performs almost the same as in the integer linear program. It also gives a better plan than the tradition approaches. We integrate our heuristic to production planning software which contains database of task steps, models and standard times. Then, it produces a schedule to suggest how the orders are processed.

Keywords: Production planning, scheduling, sofa factory.

1 Introduction

In most of factory, production planning is the important phase. It affects many things such as resource planning, cost of production, customer satisfaction etc. If the plan is not good enough, it may incur the job missing deadline which makes the late job delivery and unsatisfies the customers. Also, if the job is not planned well, the overuse or underuse of the resource may occur, i.e., overuse/underuse of the persons in the production or overuse of the material etc.

In this work, we consider a production planning algorithm in the sofa factory. We are interested in developing a software to advise the production plan. The mathematical model is used to represent the sofa production planning problem. Integer linear programming is used to solve it. Its solutions are compared to the proposed heuristic and other conventional algorithms such as FCFS, SPT and LPT. It is shown that the heuristic performs about the same as the linear programming solution and better than the other conventional ones. We integrate our algorithm in the production software. The software contains the database of the sofa models, production steps, and standard time. It shows the schedule suggesting whether the orders can be accomplished on time and if not, how the due date can be adjusted.

Many previous works exist in production planning [1-2,3,5,6,8]. David Yancey (1990) proposed FACTOR which uses for a discrete production [8]. In planning, it covers operational calendar, maintenance schedule, purchase order, etc. D. Toal et al. (2007) used database and expert systems to help support complex production processes [6]. Nowadays, there exists many uses of expert systems for scheduling a production process [5] also in a quality and production control etc. Chaimanee

(2007) developed a production scheduling in a flow production consisting of n jobs and m machines, trying to find proper starting time for each job which minimizes the total cost [1]. The problem is modeled as a linear equation system and use a linear programming to solve.

This paper is organized as follows: Section 2 presents some backgrounds about production planning and necessary mathematical model for the problem. Section 3 displays the application to our production planning problem and proposes the scheduling approach. Next, we present the mathematical model of the problem and the measurement. Then, we present the experiment data in Section 5 and Section 6 concludes the work.

2 Backgrounds

In a production, many steps are involved including: production planning, implementation and control, inventory management. In production planning, it composes of many phases: forecasting, master planning, material requirement planning [4].

In planning, we focus on scheduling which refers to planning, implying how a job is ordered, which job is executed next, what kinds of resources are needed for each job, when the output is expected. The schedule is possible under a given capacity and load.

There exists several approaches to solve the production planning such as using a chart, using linear programming, using non-linear programming, and using heuristics. To use a manual chart, it may be applicable for only a small system, containing 2-3 variables. To use a linear or non-linear system, we need to model a system of linear /non-linear equations. When we have a system containing many variables and constraints, we will model them in a set of equations. The mathematical model is needed then we solve them using a non-linear or linear programming technique. For the last method, we may develop a heuristic to select a job to schedule. The heuristic will be based on a cost function which is used to decide when and where to schedule the job. Using the mathematical models, it gives an accurate and maybe optimal solutions. But it may be difficult to model and time-consuming to solve them. Using a heuristic may be easier and give a closed-to-optimal solution under an acceptable time constraint.

Several heuristic rules exist such as following:

1. First Come First Served (FCFS) : it will schedule the job based on the arrival time.
2. Shortest Processing Time (SPT) : it will schedule the job which consumes the smallest time first.
3. Last Come First Served (LCFS): it will schedule the job which comes last first.
4. Longest Processing Time (LPT): it will schedule the job with the longest time first.
5. Earliest Due Date (EDD): It will schedule the job which contains the remaining time as the earliest due date first.

In each sofa, it contains many things such as frame, suspension system, cushion and upholstery. Each sofa model may use different material for each part. In Sofa production,

it consists of several steps as shown in Figure 1. First, we need to cut and sew (cutting and sewing). Then we perform cushion assembly. Next, in main assembly, there are two portions: main assembly 1 and main assembly2. First production contains the following steps: frame making (not including here), accessory assembly, foam assembly. In the second assembly, we need to do upholstery assembly, leg and other part assembly, cleaning and packing.

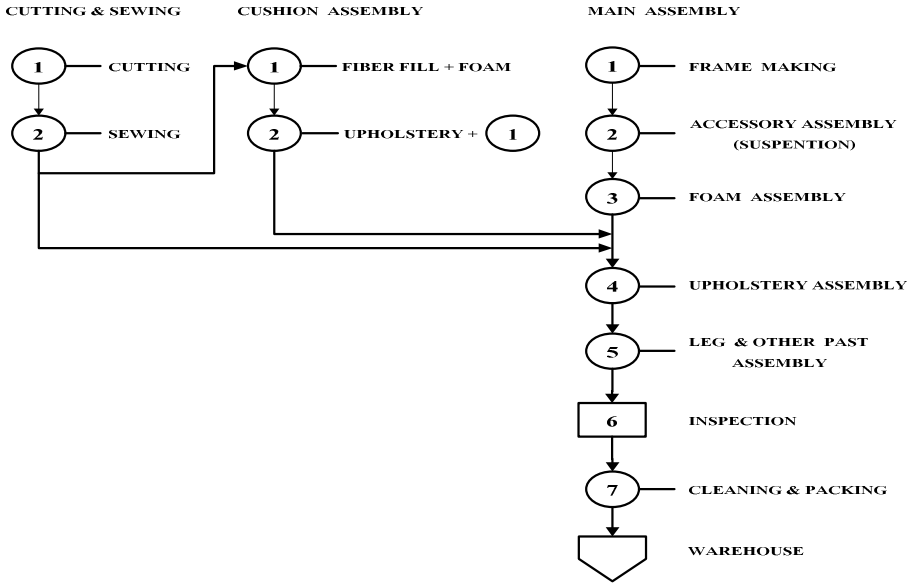


Fig. 1. Steps for a sofa production

We assume that each step use the standard time for every model in the factory. As shown in Table , it is a standard time for one model. We estimate the time by observing the worker for each step and adding the allowance time.

Table 1. Standard time for each step

No.	Steps	Time (Hours)	Process
1	cutting	0.20	Cutting & sewing
2	Sewing	0.25	Cutting & Sewing
3	Fiber fill & foam	0.25	Cushion assembly
4	Upholstery	0.30	Cushion assembly
5	Accessory assembly	0.20	Main assembly 1
6	Foam assembly	0.24	Main assembly 1
7	Upholstery assembly	0.24	Main assembly 2
8	Leg & other assembly	0.13	Main assembly 2
9	packing	0.12	Main assembly 2

In the production, the sale department will take an order from a customer, then the production plan is given. After that, the material requirement is analyzed and then the schedule is performed. For each model, we will have a table for material requirement.

3 Linear Programming Model and Measurement

For the problem, we may model the production planning assuming the following [7]:

1. Assume that total number of steps in the production is n . We know the time spent for each step, the number of jobs are totally m jobs, and the due date (for job delivery).
2. Assume that each job can not be broken down, each unit can product one job, and the efficiency of each unit is 100%.

The linear equations consist of the goal:

$$\text{Minimize } Z = \sum_{i=1}^n (E_i + T_i)$$

and the constraints are

$$C_{i,j} \geq P_{i,j} \quad \text{for } i = 1 \text{ and } j = 1 \tag{1}$$

$$C_{i,j} - P_{i,j} \geq C_{i-1,j} \quad \text{for } i = 2,3,\dots,n \text{ and } j = 1,2,\dots,m \tag{2}$$

$$C_{i,j} \geq C_{i-1,j} - P_{i-1,j} + P_{i,j} + 8 \quad \text{for } i = 1,2,\dots,n \text{ and } j = 2,3,\dots,m \tag{3}$$

$$C_{i,j} - T_i + E_i = d_i \quad \text{for } i = 2,3,\dots,n \text{ and } j = 2,3,\dots,m \tag{4}$$

where

$P_{i,j}$ is the time to process a job at position i on unit j where $i = 1,2,\dots,n$ and $j = 1,2,\dots,m$.

d_i is the due date of a job at i where $i = 1,2,\dots,n$.

$C_{i,j}$ is the finished time of the job at position i on machine unit j where $i = 1,2,\dots,n$ and $j = 1,2,\dots,m$.

E_i is the period where a job at position i finished before its due date = $\max\{d_i - C_{i,m}, 0\}$ where $i = 1,2,\dots,n$.

T_i is the time period a job at position i finished after its due date = $\max\{C_{i,m} - d_i, 0\}$ where $i = 1,2,\dots,n$.

For the first constraint, the time that a job finished at position 1 is not less than the time period to process the job. The second constraint says that the two jobs on the same unit must execute after one another. They cannot be overlapped. The third constraint says that the job on the same unit can start after the first job is finished at least 8 hours. The last constraint says that the finished time of each job on the last unit must be equal to its due date subtracted by the period that its finishes early or added by the period that its finishes late.

We measure the efficiency of the schedule using the following terms.

1. Mean flow time: it is the average flow of jobs in the system, computed by the equation.

$$F' = \frac{1}{n} * \sum_{i=1}^n F_i \tag{5}$$

where $F_i = C_i$ F_i is the flow of job i .

C_i is the time that job i is finished.

For a schedule, we want a schedule that has a small mean flow time.

2. Mean Tardiness: is the average tardiness of a job in the system.

$$T'(min) = \frac{1}{n} * \sum_{i=1}^n T_i \tag{6}$$

where $T_i = \max\{0, L_i\}$

L_i is the Time that job i finished before or after due date= $(C_i - d_i)$ if the value is negative, $L_i = 0$ which means that the job is finished before the due date. If positive, $L_i = 1$, then the job is finished after the due date

3. The number of tardy jobs: is the number of jobs that are finished after the due date. It is computed by

$$N_T = \sum_{i=1}^n \delta(T_i) \tag{7}$$

where $\delta(T_i) = 1$ when $T_i > 0$

$\delta(T_i) = 0$ when $T_i \leq 0$

Our goal also wants to minimize the number of tardy jobs.

4 Heuristic

Figure 2 shows the heuristic. First, the order is taken. Then the schedule is performed. We perform using EDD first. Then we check the schedule. If the due date is the same, we use FCFS for those. Then we check if the order date is the same, we use SPT for those. Then, we check the schedule if we can make on time, report the schedule. If not we need to adjust the due date and then the schedule is finalized.

1. Read all production orders
2. Schedule the orders using EDD
3. Check the schedule if there are tasks with the same due date. If so, use FCFS using the order date for those.
4. Check the schedule if there are tasks with the same order date. If so, use SPT for those by considering the tasks with the smallest processing time first.
5. Check the schedule if the tasks can perform according to the due dates. If so, report the schedule. If not, make a correction to the due dates. Mark the tasks overdue.

Fig. 2. Algorithm for schedules all orders

The algorithm assumes that we have the orders in prior. We test the algorithm with the real production data. The experimenopts are in the following section.

Table 2. Data Set Sample I

Order. No	Model	Type/ # seats							
		1 - seat	2 - seats	3 - seats	Single	Left- Armed	Right-armed	Corner	Stool
EQ-038/08	Arcluisis		4	23					
	Arcluisis		21	12					
EQ-040/08	Leon				11	11	11	11	
	Joy		12	12					
	Cocoon		14	17					
EQ-043/08	Arcluisis		17	17	27	18	18	18	
EQ-044/08	Arcluisis		20	20					
	Arcluisis				30	15	15	15	
EQ-045/08	Doze		17	22					
	Doze		17	21					
EQ-046/08	Joy		14						
	Toledo		70						
	Dice								50

5 Examples

We compare our heuristics to conventional ones such as FCFS, SPT, LPT in terms of mean flow time (F_i), mean tardiness time (T'), and the number of delayed jobs (NT). Also, we compare our solution to the linear programming solution. The results are presented in the following.

5.1 Sample Data Set

We took a sample data from a sofa factory. The data contains the order for one month. The data is presented in Table 2. In the table, Column “Type/#Seats” shows types of the sofa and the number of seats for a given model name (Column “Model”). Under each of these type columns, the number of order is shown.

Table 3. Time Spent for each order and schedule for sample data set 1

Tasks	Time to execute each order (day) ($P_{i,j}$)					
	038	040	043	044	045	046
CUTTING& SEWING	30	40	35	36	39	26
CUSHION ASSEMBLY	35	31	39	41	31	11
MAIN ASSEMBLY1	30	29	34	36	27	30
MAIN ASSEMBLY2	33	49	46	47	36	49
Total Time (days)	128	149	154	160	133	116
Assuming the worker works 8 hrs per day.						
Order date	5/8/08	30/8/08	5/9/08	5/9/08	5/9/08	11/8/08
Due date	17/10/08	2/10/08	27/10/08	27/10/08	3/10/08	13/10/08

Table 3 presents order information for Table 2. We pre-compute the time spent for each order according to the sofa model. The time spent here is recorded for each step of the production. This assumption is taken since we know the sofa model, the steps taken for each model and the standard time for each step accordingly. Under a grouped

Table 4. Schedule for sample data set 1

Methods	Task order					
	038	040	043	044	045	046
Heuristic	4	1	5	6	2	3
FCFS	1	2	3	4	5	6
SPT	2	4	5	6	3	1
LPT	5	3	2	1	4	6

Table 5. Comparison for all schedules for data set I

Factor	Heuristic	FCFS	SPT	LPT
F' (min)	169.83	298	231	179.17
T' (min)	0	0	0	38
N_T (min)	0	0	0	2

Table 6. Comparison for all schedules for data set II

Factor	Heuristic	FCFS	SPT	LPT
F' (min)	206.17	208	179.67	224
T' (min)	0	0	345	0
N_T (min)	0	0	2	0

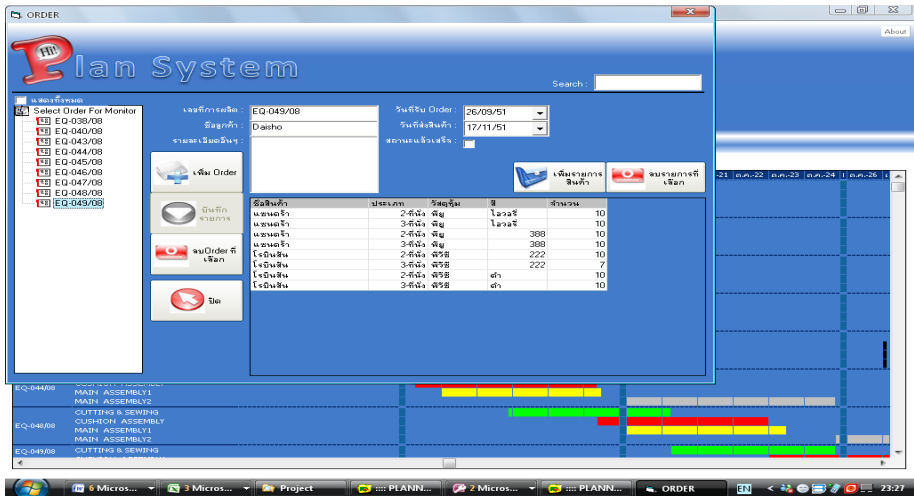


Fig. 3. Software for managing the proposed production planning

column “Time to execute each order”, there are order numbers corresponding to each row of Table 2. Under each order number column, the number of days for each order is written. Then the order date and the due dates are described for each order.

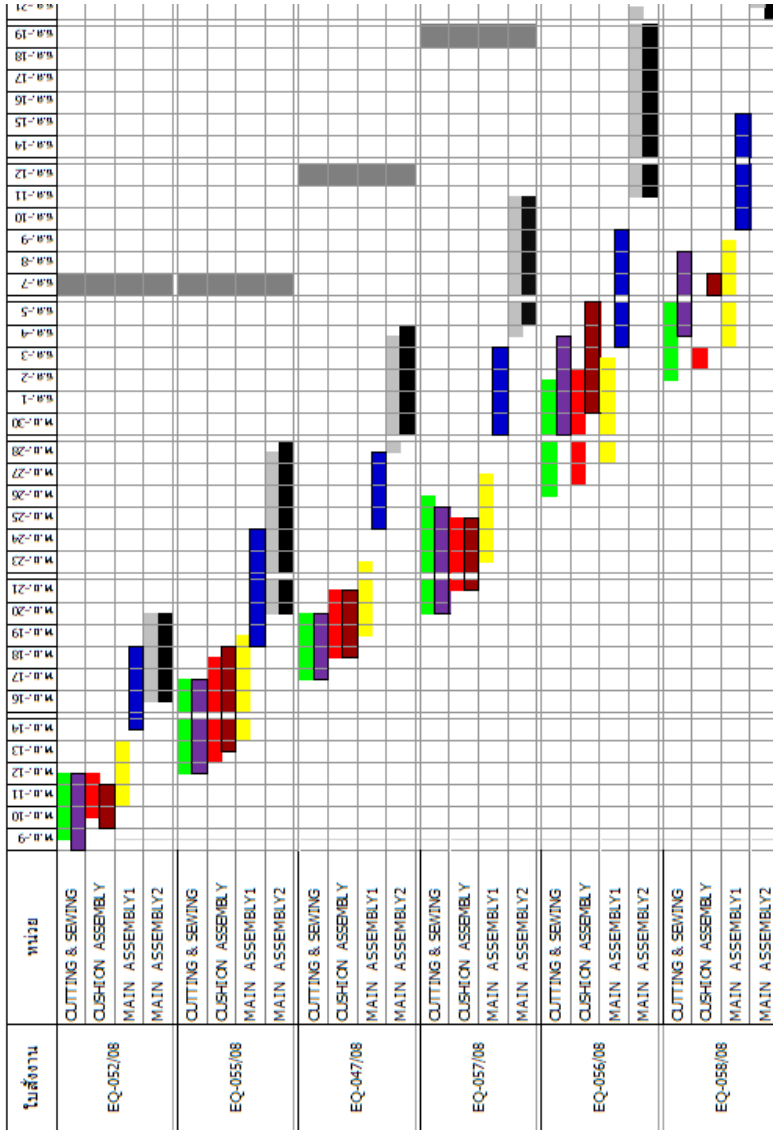


Fig. 4. Schedule compared to the real production

Table 7. Legend for Figure 4

Steps	Our heuristic	Actual
CUTTING & SEWING		
CUSHION ASSEMBLY		
MAIN ASSEMBLY1		
MAIN ASSEMBLY2		

Table 4 shows the order for each task for different approaches. Table 5 presents the comparison for the F' , T' , and N_T values. We can see that our approach has the smallest F' , while FCFS has the largest value of mean flow time. LPT has two tardy tasks. We also test again against the data set II. Though the data set is not shown here, it is shown that our heuristic give the smallest F' . Also, in this case SPT, has two tardy tasks (Table 6).

We also verify our results compared to the linear programming model (in Section 3). We used LINDO to solve the equations and found that our solution are exactly the same as in that of the linear programming solution. Hence, from the experiments, it is shown that the proposed heuristic performs quite well.

We implement our algorithm in the planning software in Figure 3. The software contains database for each sofa model and type including standard time for each step. After it takes order information, it suggests the schedule based on our algorithm. It shows the schedule suggesting whether the orders can be accomplished on time and if not, how the due date can be adjusted.

Figure 4 compares the schedule produced by our software to the real production. Table 10 shows legends for each step for each approach. We can see that the proposed schedule is still optimistic. In reality, factors such as the readiness of the task, should be considered. This will be added in the further. In the actual production, there exists a problem of availability of resources such as human labor, materials which makes the tasks even delay.

6 Conclusion

We proposed a heuristic and software for sofa production planning. Our heuristic considers the due date, the order date, and the processing time. We compared to the conventional approach which considers only each factor such as EDD, SPT, LPT. It is found that our heuristic performs better. Also, we verify the solution obtained by our heuristic with the integer linear programming model. It is seen that our solutions are closed to the linear programming solution. We integrate our algorithm into the production planning software. The software has a database for the models, types of sofa, steps to performs and standard time. It computes the schedule using our heuristic. We compare our schedule with the actual schedule in the production. It is seen that our schedule is still optimistic since the readiness of the tasks are not yet considered. If so, the schedule will be more realistic. This will be considered in the software in the future.

References

- [1] Chaimane, et al.: Specification of task starting time for flow production in real-time. In: Proceedings of NCRT, pp. 113–125 (2007)
- [2] Kops, L., Natarajan, S.: Time partitioning based on the job-flow schedule — A new approach to optimum allocation of jobs on machine tools. *International Journal of Advanced Manufacturing Technology* 9(3), 204–210 (1994)
- [3] Ipsilandis, P.G.: Multiobjective Linear Programming Model for Scheduling Linear Repetitive Projects. *Journal of Construction Engineering and Management* 133(6), 417–424 (2007)
- [4] Pinedo, M.L.: *Scheduling: Theory, Algorithms, and Systems*, 3rd edn. Springer, Heidelberg (2008)
- [5] Schmidt, G.: Two-machine n-job flow-shop scheduling problem with limited machine availability. In: Proceedings of 14th Workshop on Discrete Optimization, TU Freiberg (2000)
- [6] Toal, D., Coffey, T., Smith, P.: *Expert Systems and Simulation in Scheduling*. <http://www.ul.ie/~toald/Publications/lmc-11es.pdf> (Accessed: February 20, 2007)
- [7] Winston, W.L.: *Operations Research: Applications and Algorithms*, 3rd edn. Duxbury Press, Boston (1994)
- [8] Yancey, D.P.: Implementation of rule-based technology in a shop scheduling. In: Proceedings of the 3rd international conference on Industrial and engineering application of artificial intelligence and expert system, IEA/AIE, vol. 1, pp. 865–873 (1990)

Ensemble Learning for Imbalanced E-commerce Transaction Anomaly Classification

Haiqin Yang and Irwin King

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, New Territories, Hong Kong
{hgyang, king}@cse.cuhk.edu.hk

Abstract. This paper presents the main results of our on-going work, one month before the deadline, on the 2009 UC San Diego data mining contest. The tasks of the contest are to rank the samples in two e-commerce transaction anomaly datasets according to the probability each sample has a positive label. The performance is evaluated by the lift at 20% on the probability of the two datasets. A main difficulty for the tasks is that the data is highly imbalanced, only about 2% of data are labeled as positive, for both tasks. We first preprocess the data on the categorical features and normalize all the features. Here, we present our initial results on several popular classifiers, including Support Vector Machines, Neural Networks, AdaBoosts, and Logistic Regression. The objective is to get benchmark results of these classifiers without much modification, so it will help us to select a classifier for future tuning. Further, based on these results, we observe that the area under the ROC curve (AUC) is a good indicator to improve the lift score, we then propose an ensemble method to combine the above classifiers aiming at optimizing the AUC score and obtain significant better results. We also discuss with some treatment on the imbalance data in the experiment.

1 Introduction

The 2009 UC San Diego data mining contest is a yearly competition for undergraduate students, graduate students, and postdoctoral researchers in colleges since 2004. The goal of this year's contest is to design computational methods to rank the example in the two datasets, where data are from anomalous web transactions, according to the probability each example has a positive label. The following is a description of the data and we summarize them in Table 1.

- The contest consists of two tasks, one is named “easy” and the other is named “hard”. They are two datasets involving 19 features from web transaction anomaly data, where two features are the state and email information of the transaction and the other 17 features can be deemed as continuous features. The features corresponding to state and email information are categorical features.

Table 1. Data Description

Task	# Feature	Train			Test
		Total	Positive	Negative	
Easy	19	94,682	2,094	92,588	36,019
Hard	19	100,000	2,654	97,346	50,000

- For the task 1 (the “easy” task), the training data consist of 94,682 examples and the test set consists of 36,019 examples. The test set is drawn from the same distribution as the training set.
- For the task 2 (the “hard” task), the training data consist of 100,000 examples and the test set consists of 50,000 examples, where the test set is drawn from the same distribution as the training set.

There are difficulties encountered from the distribution of the data as well as the evaluation criterion:

- The class distribution in the datasets is highly imbalanced. There are roughly fifty times as many negative examples as positive. In this case, standard classifiers tend to have a bias in favor of the larger classes and ignore the smaller ones.
- The evaluation criterion is lift at 20% on the probability each example has a positive label of the datasets. That is, it takes the first 20% of the sorted list of predicted values with the biggest values, and makes a list of the original indices of this top 20%. The result counts the number of true positives in the list. That means a perfect classifier can get the best score as 5. The evaluation is different to the objective in standard classifiers, which aim at optimizing the error rate.

Based on the data characteristics and specific evaluation criterion, we first preprocess the data on the categorical features and normalize them. Here, we present the results of our first stage testing. Hence, our objective is to test on several popular classifiers, including Support Vector Machines (SVMs), Neural Networks (NNs), AdaBoosts, and Logistic Regression, to get their benchmark results, without much modification. These basic results can be used to choose a better classifier for further tuning. Further, after observing that the area under the ROC curve (AUC) [3,5] is a good indicator to improve the test performance, we then ensemble the above four classifiers to get a powerful classifier by optimizing the AUC score. Significant better results are obtained on both tasks.

The rest of this paper is organized as followings: In Section 2, we illustrate the test procedure and describe the methodologies adopted. In Section 3, we detail the procedure of parameter seeking on the models, the current results, observations, and our other testing. Finally, we conclude the paper in Section 4.

2 Flow and Methodologies

The test consists of three main processing steps: 1) data preprocessing, including categorical features preprocessing and data normalization; 2) classifiers building with parameters tuning and models ensemble; 3) output of test results: the probability of each sample being assigned to positive label. In the following subsections, we will describe the above procedure in details.

2.1 Data Preprocessing

The data consist of 17 continuous features and two categorical features containing the state and the email information. For the state feature, there are 54 states in the transaction datasets. Most transactions are recorded the state as “CA” and some states, e.g., “AE”, “AP”, etc. only appear in several transactions. Hence, we categorize the state feature based on the number of the transaction happened on the state. Concretely, we first set the state as a specific category when the number of the transaction on that state is in one digit order. Next, we set the state into a new category based on the number of transactions in the order of each 100, each 1,000, and 10,000. After that, we obtain 20 and 17 categories to represent all 54 states in the state feature for the “easy” task and the “hard” task, respectively. We then expand the state feature into a 20-dimensional and 17-dimensional features with 1 indicating the corresponding categorized state and 0 when the state does not appear in that transaction and that category.

In the email feature, some domains, e.g., ‘AOL.COM’, ‘COMCAST.NET’, etc., appear frequently. Other domains, e.g., ‘.MIL’, etc., seldom appear in the transactions. Similarly, based on the frequency of domains appearing in the transactions, we categorize the email domains into 19 types and expand the email feature into 19-dimensional features.

After concatenating the continuous features with expanded state features and expanded email features, we obtain the corresponding training data and test data. We further use a standard method to normalize them: making the sum square for each feature in the training data to 1, and normalize the test data according to the weight in the training data. Due to the number of features is relative small comparing to the number of training data, we do not perform feature selection on both tasks further.

2.2 Classifiers

In the test, we first explore several popular classifiers, including Support Vector Machines (SVMs) [21], Adaboost [19], Neural Networks [2], and Logistic Regression [8], to get their benchmark results. These results are used to select a better classifier for further tuning.

Support Vector Machines. Highly imbalance of the data is a major difficulty in the contest. To solve the imbalance problem, in SVM, we seek a decision

boundary, $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$, where $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$, by adding different weights on the cost of different label of data as follows:

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

where training data are $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, with N instance-label pairs. C^+ and C^- are weights of training errors with respect to the positive and negative samples, respectively. In the test, we set C^+ to be 50 times larger than C^- . Since the task is in large scale, we just seek linear classifier for SVMs. The output scores are then re-scaled by $1/(1 + \exp(-f(\mathbf{x})))$.

Here, we adopt a very basic routine on the SVMs to get benchmark results. Other methods, e.g., support vector method for the AUC score [12] may be adopted to get better performance; probability output of svms [17] may be adopted to fit the evaluation metric of the task.

Neural Networks. Neural networks (NNs), also called artificial neural networks, are computational models that can capture the non-linear property or structural relation embedded in the data [2]. Their powerful computation ability motivates us to test the performance on the tasks. Here, we adopt a radial basis function (RBF) network, which uses radial basis functions as activation functions and combines these radial basis functions in a linear form. Here, we use an implementation of a RBF network in [16]. The drawbacks of the RBF network are that there are some parameters need to be tuned and the model is easy to seek a local optimal solution.

Adaboost. Boosting is a very efficient and effective method to find a classification rule by combining many “weak” learners, in particular when each of which is only moderately accurate. In the test, we also tried the AdaBoost [19].

The main idea of AdaBoost is to construct a highly accurate classifier by combining many weak learners. The weak learners are only moderately accurate but should be diverse. Currently, there are some extensions or generalizations from the basic AdaBoost algorithm first introduced by Freund and Schapire [6]. These extensions include the Real AdaBoost [19], the Modest Adaboost [22], and etc.

Here, we choose Real AdaBoost [19] implemented by [20], which supports real-value prediction and obtains better performance. The weak learner we used is classification and regression tree (CART). This is because CART is inherently suited for imbalanced dataset since its tree is constructed according to the correct classified ratio of positive and negative examples and the model selection procedure can be done simply and efficiently by iteratively increasing the number of weak learners and stopping when the generalization ability on the validation set does not improve. In the test, we change the number of splits in the CART and the number of iterations for the Real Adaboost to get a better benchmark result.

Logistic Regression. Logistic regression (LR) [8] is a standard tool to predict the probability of occurrence of an event by fitting data to a logistic curve. It can output the probability directly, which exactly fit the evaluation criterion of the contest. The output of logistic regression is a probability in the following form:

$$P_{LR} = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}}, \quad (2)$$

where the parameters \mathbf{w} and b are estimated by maximum likelihood. The advantage of the logistic regression is that there is no parameter to be tuned and the model can achieve relative better results.

2.3 Models Ensemble

Literature states that combining divergent but fairly high performance models into an ensemble can usually lead to a better generalization performance [7,13,14]. Other than combining divergency, ensemble method may also play the role of voting to help the generalization performance [18]. Here, we combine the output of the above four base classifiers in a linear form as follows:

$$P_{final} = w_1 * P_{SVM} + w_2 * P_{RBF} + w_3 * P_{Adaboost} + w_4 * P_{LR} \quad (3)$$

The above weights, $w_i, i = 1, \dots, 4$, are selected uniformly from $[0, 1]$ to tune a powerful ensemble classifier. The voting scheme is then incorporated by the values of the weights. Large value in the corresponding weight means that it votes towards the result of the corresponding classifier.

From the preliminary results on individual classifiers, we notice that the test performance, or the lift score, is proportional to the AUC score. A higher AUC score on the training data corresponds to a higher lift score on the test data. Hence, in tuning the ensemble model, we seek to optimize the AUC score of the model. Since w_i can be set to 0, some models will be automatically discarded when seeking a better ensemble model.

3 Experiments and Current Results

In the test, we first test the basic performance of individual models. In order to quickly obtain preliminary results, we use different training size on the models. More specifically, we randomly split the training data into 10 folds, where nine folds are used for training the SVMs, RBF nets, and Logistic Regression, and the rest fold is used to test the AUC score of these models. For the Real AdaBoost, we use only one twentieth of the training data in the training procedure and use the rest for test, due to the computation consideration. Since only one submission is allowed in one day for the contest and we observe that the AUC score is a good indicator to attain better test result, we apply the trained classifier corresponding to highest AUC score in each individual model for the test data to get the lift score. In the following, we detail the parameters seeking in different models:

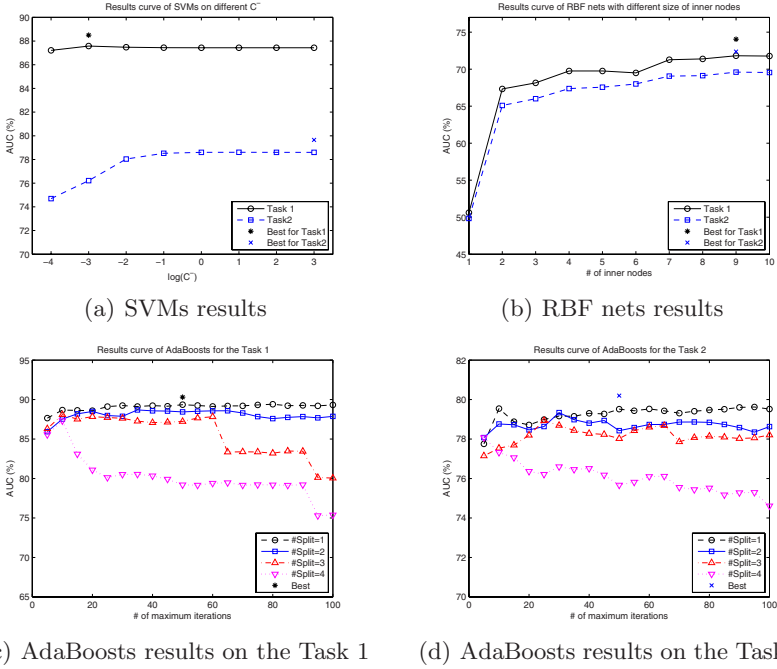


Fig. 1. Validation results curve on the training data

SVMs: an SVM implemented by LibSVM [4] with linear kernel is adopted due to the computational consideration. The parameter of C^- in SVMs is tested from 10^{-4} to 10^3 and C^+ is set to 50 times larger than C^- .

AdaBoost: the Real AdaBoost implemented by [20] are tested with different number of splits in the tree and different number of maximum iteration. The number of splits in the tree is enumerated from 1 to 4 and the maximum iteration is tested from 5 to 100 with each step being 5.

NNs: RBF nets implemented by [16] are tested the number of inner nodes from 1 to 10 with other parameters being default.

Logistic Regression: no parameters need to be set.

Ensemble Model: weights, $w_i, i = 1, \dots, 4$, are selected uniformly from $[0, 1]$ to tune a powerful ensemble classifier on the obtained above best models.

We show the results of individual models in Fig. 1 and report the best results obtained for all models in Table 2. From results, we have the following observations:

- The parameter C in SVMs is less sensitive to the task 1 and less sensitive to the task 2 when C is large.
- The AUC scores increase as the number of inner nodes increases for RBF nets and become less sensitive when the number of inner nodes is large.

Table 2. Results on different models. Lift scores are obtained when uploaded the results on the test sets. AUC scores are results on the inner test sets.

Method	Easy		Hard	
	AUC (%)	Lift	AUC (%)	Lift
SVM	88.5	3.699	79.7	2.771
RBF	74.0	1.964	72.4	2.664
AdaBoost	90.3	3.826	81.1	3.024
LR	90.1	3.82	80.1	2.984
Ensemble	92.0	4.235	82.2	3.115

- The AUC scores decrease as the number of split nodes increases for both tasks. They attain the maximum scores when the number of iterations equals 50 for both tasks.
- For individual models, Adaboost obtains the best lift score and AUC score for both tasks while Logistic Regression attains the second best lift score and AUC score. A higher AUC score on the test result of the training dataset corresponds to a higher lift score obtained from the submitted results.
- After obtained the ensemble model, we obtain a significant improvement on both AUC score and lift score for both tasks, which are the best among all the models. The experimental results indicate that the ensemble model works like as a voting scheme: a model with better performance has a larger weight.

In the above, we report the preliminary results on the contest. Since the number of given features is relative small, there may be non-linearity embedded in the data, we also use some techniques, e.g., spline [23], to expand the features and achieve better results. Finally, we find that a bottleneck is the highly imbalance in the data. Usually, standard classifiers tend to bias in favor of the larger class since by doing so it can reach high classification accuracy. Researchers usually adopt methods such as down-sampling of major class, up-sampling of minor class, or class-sensitive loss function, to tackle the imbalance data problem [15,24]. A more systematic method, the Biased Minimax Probability Machine, to solve the imbalance data problem is also proposed in the literature [11,10,9]. Due to computation consideration, for Adaboost, we modify the model by adjusting the dependent variable [1]. For other methods, we adopt a standard method, down-sampling on the negative samples, to alleviate the imbalance problem. However, the above methods are in heuristic way and data dependent. Seeking good parameters or good sub-samples is time consuming and we do not find much improvement on it. After trying several other methods, we can improve the lift score to 4.26 for the “easy” task and 3.19 for the “hard” task. Our results are ranked in top 20 for the “easy” task and top 10 for the “hard” task in one month before the deadline.

4 Conclusions

In this paper, we summarize our on-going work on the 2009 UC San Diego data mining contest. We have preprocessed the categorical features and tested on several standard classifiers, e.g., SVMs, RBF nets, AdaBoost, and Logistic Regression, without much modification, to get the preliminary results. The results reported in this stage can be used as reference to select classifiers for further tuning. Further, after notice that the AUC score on the training data is a good indicator to improve the lift score on the test data, we propose an ensemble model to optimize the AUC score and achieve significant better results.

References

1. Bell, R.M., Haffner, P.G., Volinsky, J.C.: Modifying boosted trees to improve performance on task 1 of the 2006 kdd challenge cup. *ACM SIGKDD Explorations Newsletter* 2, 47–52 (2006)
2. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford (1996)
3. Bradley, A.: The use of the area under the ROC curve in the evaluation of machine learning algorithm. *Pattern Recognition* 30(7), 1145–1159 (1997)
4. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
5. Fawcett, T.: An introduction to roc analysis. *Pattern Recognition Letters* 27, 861–874 (2006)
6. Freund, Y., Schapire, R.E.: Game theory, on-line prediction and boosting. In: *Proc. of the Ninth Annual Conference on Computational Learning Theory*, pp. 325–332 (1996)
7. García-Pedrajas, N., García-Osorio, C., Fyfe, C.: Nonlinear boosting projections for ensemble construction. *Journal of Machine Learning Research* 8, 1–33 (2007)
8. Hosmer, D.W., Lemeshow, S.: *Applied logistic regression*, 2nd edn. Wiley-Interscience Publication, Hoboken (2000)
9. Huang, K., Yang, H., King, I., Lyu, M.R.: Imbalanced learning with biased minimax probability machine. *IEEE Transactions on System, Man, and Cybernetics Part B* 36, 913–923 (2006)
10. Huang, K., Yang, H., King, I., Lyu, M.R.: Maximizing sensitivity in medical diagnosis using biased minimax probability machine. *IEEE Transactions on Biomedical Engineering* 53, 821–831 (2006)
11. Huang, K., Yang, H., King, I., Lyu, M.R., Chan, L.: The minimum error minimax probability machine. *Journal of Machine Learning Research* 5, 1253–1286 (2004)
12. Joachims, T.: A support vector method for multivariate performance measures. In: *ICML*, pp. 377–384 (2005)
13. Juditsky, A., Rigollet, P., Tsybakov, A.B.: Learning by mirror averaging. *Annals of Statistics* 36, 2183–2206 (2008)
14. Kégl, B., Busa-Fekete, R.: Boosting products of base classifiers. In: *ICML*, p. 63 (2009)
15. Maloof, M.A., Langley, P., Binford, T.O., Nevatia, R., Sage, S.: Improved rooftop detection in aerial images with machine learning. *Machine Learning* 53, 157–191 (2003)

16. Nabney, I.T.: Netlab: Algorithms for Pattern Recognition. Springer, Heidelberg (2004)
17. Platt, J.C., Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: *Advances in Large Margin Classifiers*, pp. 61–74. MIT Press, Cambridge (1999)
18. Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.S.: Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics* 26, 1651–1686 (1998)
19. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. *Machine Learning* 37(3), 297–336 (1999)
20. G.A.M. Toolbox, <http://research.graphicon.ru/machine-learning/gml-adaboost-matlab-toolbox.html>
21. Vapnik, V.: *The Nature of Statistical Learning Theory*, 2nd edn. Springer, New York (1999)
22. Vezhnevets, A., Vezhnevets, V.: Modest adaboost – teaching adaboost to generalize better. Graphicon (2005)
23. Wahba, G.: *Spline Models for Observational Data*, volume 59. In: *CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 59. SIAM, Philadelphia (1990)
24. Weiss, G.M.: Mining with rarity: a unifying framework. *SIGKDD Explorations* 6(1), 7–19 (2004)

Exploring Early Classification Strategies of Streaming Data with Delayed Attributes

Mónica Millán-Giraldo, J. Salvador Sánchez, and V. Javier Traver

Dept. Llenguatges i Sistemes Informàtics
Universitat Jaume I
Av. Sos Baynat s/n, 12071 Castelló de la Plana, Spain

Abstract. In contrast to traditional machine learning algorithms, where all data are available in batch mode, the new paradigm of streaming data poses additional difficulties, since data samples arrive in a sequence and many hard decisions have to be made on-line. The problem addressed here consists of classifying streaming data which not only are unlabeled, but also have a number l of attributes arriving after some time delay τ . In this context, the main issues are what to do when the unlabeled incomplete samples and, later on, their missing attributes arrive; when and how to classify these incoming samples; and when and how to update the training set. Three different strategies (for $l = 1$ and constant τ) are explored and evaluated in terms of the accumulated classification error. The results reveal that the proposed on-line strategies, despite their simplicity, may outperform classifiers using only the original, labeled-and-complete samples as a fixed training set. In other words, learning is possible by properly tapping into the unlabeled, incomplete samples, and their delayed attributes. The many research issues identified include a better understanding of the link between the inherent properties of the data set and the design of the most suitable on-line classification strategy.

Keywords: Data mining, Streaming data, On-line classification, Missing attributes.

1 Introduction

Most of traditional learning algorithms assume the availability of a training set of labeled objects (examples or instances) in memory. In recent years, however, advances in information technology have led to a variety of applications in which huge volumes of data are collected continuously, thus making impossible to store all data, or process any particular object more than once. Under these circumstances, data are not available as a batch but comes one object at a time (called *streaming data*). In general, a data stream is defined as a sequence of instances [2, 11]. Data streams differ from the conventional model in important elements [4] that bring new challenges: (i) The objects in the stream arrive on-line; (ii) The system has no control over the order in which incoming data arrive to be processed; (iii) Data streams are potentially unbounded in size.

Classification is perhaps the most widely studied problem in the context of data stream mining. Although substantial progress has been made on this topic [1, 6, 14], a number of issues still remain open. For example, many classification models do not make adequate

use of the history of data streams in order to accommodate changes in class distribution (known as *concept drift* [8,15,16]). The scenario we consider in this paper faces a new problem that may appear in several real-world applications. We assume that each object of a data stream is a vector of d attribute values without a class label. The aim of the classification model is to predict the true class of each incoming object as soon as possible (ideally, in real time). However, suppose that the attribute values are obtained from different sensors. These may produce that the attribute values will become available at different times if some sensor requires more processing time to compute an attribute value than the others or even, if some sensors fail. Therefore we are considering the problem of classifying streaming data where one or more attributes arrive with a delay. As an example, when a sensor fails in a production process, it might not be feasible to stop everything and in this case, the system should employ the information available at present time. Three main issues here are: (i) How to classify the incoming sample with missing attributes; (ii) Whether to update the training (reference) set after predicting the class label of an incomplete object or wait until the attribute vector has been completed; and (iii) What to do when the missing attributes arrive.

In the literature, there exist many algorithms for handling data with missing attributes in off-line learning [7,9,10], but no one is absolutely better than the others. The most representative categories of these are:

1. Removing examples with missing attributes: The simplest way of dealing with missing values is to discard the examples that contain the missing values. This technique may lose relevant information.
2. Projection: The l missing attributes are ignored. This implies to map the d dimensional input vectors onto an $(d - l)$ instance space.
3. Imputation: It tries to guess the missing values. In fact, usually missing values depend on other values, and if we find a correlation between two attributes, we may use it to impute missing items. Imputations may be deterministic or random (stochastic). In the first case, imputations are determined by using the complete data, and are the same if the method is applied again. In the second case, imputations are randomly drawn.

Despite the problem of missing attributes has been widely studied in off-line learning, to the best of our knowledge it has not previously been considered in the context of on-line learning with streaming data, which makes the problem considerably more challenging. This paper reports a preliminary study of three straightforward strategies for an early classification of streaming data with missing attributes. By *early* we mean that classification of an incoming object is done *before* the whole attribute vector is known. Many applications can benefit from performing this early classification, since there may be some kind of loss associated with waiting for the missing attributes to arrive. In the present work we concentrate on the case of a single missing attribute which happens to be the same and arrive with a constant delay.

2 Classification of Streaming Data with Delayed Attributes

At time step t in the scenario of attributes arriving with a delay, we have a reference set S_t (a set of labeled examples with all attributes available). Then, a new unlabeled

object \mathbf{x}_{t+1} with one missing attribute $x_{t+1}^{(i)}$ arrives. After predicting the label for \mathbf{x}_{t+1} , the system receives the value of the attribute $x_{t-\tau+1}^{(i)}$ corresponding to the object that came τ steps earlier, $\mathbf{x}_{t-\tau+1}$. Therefore, objects from $\mathbf{x}_{t-\tau+2}$ to \mathbf{x}_{t+1} are still with one missing attribute.

Here, one key question is whether to use the unlabeled data with missing attributes to update the reference set S_t and in such a case, how to do it. In addition, we have to decide how to best utilize the value of the missing attribute $x_t^{(i)}$ when this arrives.

When a new unlabeled object \mathbf{x}_{t+1} arrives, the system has to provide a prediction for its label based on the information available up to time t . In this situation, it would be desirable to make use of the *confidence* with which the previous classifications have been made. That is why a modification of the k -Nearest Neighbors (k -NN) rule [17] is here used, since its stochastic nature results suitable to properly manage the confidence measurements. On the other hand, for handling the missing attribute of object \mathbf{x}_{t+1} , we employ the projection strategy because of its simplicity and its proven good behavior.

2.1 A Classifier with Confidence Measurements

All instances in the reference set have a confidence value for each class, indicating the probability of belonging to the corresponding class. When a new unlabeled object \mathbf{x}_{t+1} from the data stream arrives, its confidence values (one per class) are estimated. Thus the object will be assigned to the class with the highest confidence value.

To estimate the confidence values of the incoming object \mathbf{x}_{t+1} , its k nearest neighbors (NN) from the reference set S_t are used. The confidences of its k nearest neighbors, which contribute a weight by each class to the object \mathbf{x}_{t+1} , and the distances between them and the new object \mathbf{x}_{t+1} are also employed. More formally, let k be the number of nearest neighbors, let \mathbf{n}_j be the j -th nearest neighbor of \mathbf{x}_{t+1} , let $p_m(\mathbf{n}_j)$ denote the confidence (probability) that the j -th nearest neighbor belongs to class m , and let $d(\mathbf{x}_{t+1}, \mathbf{n}_j)$ be the Euclidean distance between the object \mathbf{x}_{t+1} and \mathbf{n}_j . The confidence of the object \mathbf{x}_{t+1} in relation with the class m , say $P_m(\mathbf{x}_{t+1})$, is given by:

$$P_m(\mathbf{x}_{t+1}) = \sum_{j=1}^k p_m(\mathbf{n}_j) \frac{1}{\epsilon + d(\mathbf{x}_{t+1}, \mathbf{n}_j)}, \quad (1)$$

where ϵ is a constant value ($\epsilon = 1$), which is employed to avoid uncertain values in the division when the object \mathbf{x}_{t+1} is very similar or very close to its j -th nearest neighbor.

The above expression states that the confidence that an object \mathbf{x}_{t+1} belongs to a class m is the weighted average of the confidences that its k nearest neighbors belong to class m . The weight is inversely proportional to the distance from the object to the corresponding k nearest neighbors. In order to get a proper probability, the confidence $P_m(\mathbf{x}_{t+1})$ in Eq. (1) is divided by the sum of the confidences of the k nearest neighbors to all the c classes:

$$p_m(\mathbf{x}_{t+1}) = \frac{P_m(\mathbf{x}_{t+1})}{\sum_{r=1}^c P_r(\mathbf{x}_{t+1})}, \quad (2)$$

As the objects of the reference set S_t are labeled elements, their confidence values were initially set to 1 for the true class (the class to which they belonged), and zero for the

remaining classes. During the on-line learning, the confidence of all new objects added into the training set will be updated according to the probability values of Eq. (2).

2.2 Managing Incomplete Objects and Their Delayed Attribute

Assuming that at step t we have a reference set S_t available, on-line classification of incomplete data streams consists of three elements: (i) The technique to handle the situation of a missing attribute $x_{t+1}^{(i)}$ of the new unlabeled object \mathbf{x}_{t+1} ; (ii) The classifier to predict the class label for this object; and (iii) The strategy to manage the new information derived from the value of the attribute $x_{t+1}^{(i)}$ when it arrives τ steps later.

Regarding the first issue, as stated before, the projection strategy is used: the arriving object as well as those in the reference set are simply mapped onto the $d-1$ dimensional space. Second, as for the prediction of the class label for \mathbf{x}_{t+1} , the k -NN classifier based on posterior probabilities (Sect. 2.1), is used. Finally, since it is not obvious which is the best way to profit from the new information gained with the arrival of the attribute $x_{t-\tau+1}^{(i)}$ at time step $t+1$, three different strategies are explored:

1. *Do-nothing*: This is a *passive* strategy where, while the incoming object is incorporated into the current reference set S_t , nothing is done when the value of the missing attribute $x_{t-\tau+1}^{(i)}$ arrives after τ time steps. However, the attribute value of the corresponding object, $\mathbf{x}_{t-\tau+1}$, is set to the value $x_{t-\tau+1}^{(i)}$.
2. *Put-and-reclassify*: This is a *proactive* strategy differing from the *do-nothing* strategy in that the object $\mathbf{x}_{t-\tau+1}$ is also reclassified, this time using *all* attributes.
3. *Wait-and-classify*: This is a *reactive* strategy where, unlike the two previous strategies, the new object \mathbf{x}_{t+1} is *not* included in the reference set S_t until its missing attribute is received after τ time steps. Only by then, the complete object is classified and incorporated into the reference set $S_{t+1+\tau}$.

The different nature of these strategies will allow to gain some insight into which may be the best way to proceed in the context of on-line classification of streaming data with missing (but delayed) attributes. This will also provide cues on what further research avenues to follow. The assessment of the different strategies proposed has been done on extensive experimental work, which is subsequently presented.

3 Experiments and Results

The experiments here carried out are directed to empirically evaluate each strategy described in the previous section, pursuing to determine which of these is the most suitable for the classification of incomplete streaming data. The ultimate purpose of this preliminary study is to investigate whether the employment of attribute values that arrive with a delay allows to improve the system performance or not.

Experiments were conducted as follows:

Data sets: Fifteen real data sets (summary of whom is given in Table 1) were employed in the experiment. Data were normalized in the range $[0, 1]$ and all features were numerical. In the table, the data sets are sorted by increasing size.

Table 1. Characteristics of the real data sets used in the experiments

Data set	Features	Classes	Objects	Reference Set	Data Stream	Source
iris	4	3	150	12	138	UCI ¹
crabs	6	2	200	12	188	Ripley ²
sonar	60	2	208	120	88	UCI
laryngeal1	16	2	213	32	181	Library ³
thyroid	5	3	215	15	200	UCI
intubation	17	2	302	34	268	Library
ecoli	7	8	336	56	280	UCI
liver	6	2	345	12	333	UCI
wbc	30	2	569	60	509	UCI
laryngeal2	16	2	692	32	660	Library
pima	8	2	768	16	752	UCI
vehicle	18	4	846	72	774	UCI
vowel	11	10	990	110	880	UCI
german	24	2	1000	48	952	UCI
image	19	7	2310	133	2177	UCI

¹UCI [3]²Ripley [12]³Library http://www.bangor.ac.uk/~mas00a/activities/real_data.htm

Partitions: For each database, 10 runs were carried out. A random stratified sample of $d \times c$, being d the number of attributes and c the number of classes, was taken as the initial labeled reference set S_0 . The remaining part of each database was used as the incoming on-line streaming data. To simulate independent and identically-distributed sequences, the data were shuffled before each of these 10 runs.

Incomplete objects: A new object with one missing attribute from the on-line data was fed to the system at a time step. Both the most and the least relevant attributes of each database were simulated to be missing. Attribute relevance was estimated by means of the Jeffries-Matusita distance [5].

Delay: The value of the missing attribute comes after $\tau = 5$ time steps. When the delayed attribute arrives, the corresponding object is completed with the true attribute value.

Classification: At each time step t , the respective strategy to handle delayed attributes was applied. The accumulated classification error (the total number of misclassifications divided by the number of samples processed up to t) was computed. In this way we created a progression curve (trend line), which is the classification error as a function of the number of on-line objects seen by the classifier. The results were averaged across the 10 runs giving a single progression curve for each data set.

For each of the 10 runs of the experiment, all strategies received the same partitions of the data into initial labeled reference set and streaming data set. These on-line data were presented to all methods in the same order so that performance differences can not be attributable to different data (order).

3.1 Results

Table 2 reports the average errors estimated across all incoming objects for each strategy. To evaluate whether the performance improves at all with streaming data, we have also included the error using only the initial reference set S_0 . For each database, the first row corresponds to the classification errors when the least relevant attribute arrives with a delay. The second row is for the most relevant attribute. Highlighted in bold are the results being better than those obtained by using only the initial reference set for classification. Underlined values correspond to the best strategy for each database and each attribute. As can be seen, out of the 15 databases, the strategies here proposed give better results than using the initial reference set on 14 cases when the missing attribute

Table 2. Average errors estimated across all incoming objects

Data set	Initial reference set	Do-nothing	Put-and-reclassify	Wait-and-classify
iris	0.1076	0.0954	0.0919	0.0893
	0.1236	0.1134	0.1111	0.1093
crabs	0.4695	0.4332	0.4226	0.4313
	0.4875	0.4341	0.4334	0.4456
sonar	0.2299	0.2205	0.2206	0.2207
	0.2319	0.2225	0.2223	0.2239
laryngeal1	<u>0.1974</u>	0.2033	0.2044	0.2051
	0.1931	0.1915	0.1915	0.1957
thyroid	0.1860	0.1639	0.1751	0.1763
	0.2531	0.2336	0.2158	0.2168
intubation	<u>0.3214</u>	0.3714	0.3631	0.3655
	<u>0.3581</u>	0.3899	0.3859	0.3838
ecoli	<u>0.2060</u>	0.2146	0.2115	0.2120
	0.1834	0.1841	0.1786	0.1782
liver	0.4689	0.4610	0.4607	0.4648
	0.4732	0.4686	0.4631	0.4695
wbc	0.0427	0.0417	0.0414	0.0423
	0.0467	0.0432	0.0411	0.0428
laryngeal2	0.0545	0.0538	0.0533	0.0526
	0.0538	0.0526	0.0524	0.0516
pima	<u>0.3120</u>	0.3290	0.3271	0.3287
	0.3415	0.3383	0.3353	0.3400
vehicle	0.4451	0.4336	0.4334	0.4368
	0.4383	0.4366	0.4337	0.4339
vowel	0.4410	0.4180	0.4162	0.4169
	0.4960	0.4608	0.4530	0.4541
german	<u>0.3156</u>	0.3225	0.3216	0.3202
	0.3367	0.3434	0.3350	0.3334
image	0.1204	0.1117	0.1169	0.1166
	0.1410	0.1269	0.1254	0.1251

corresponds to the most relevant, and on 10 for the least relevant attribute. Performance differences are small among the proposed strategies as well as between each of them and the baseline case.

Detailed results for two databases are provided in Fig. 1 with the x and y axes representing, respectively, the number of objects fed to the system at each time step, and the accumulated classification error averaged over the 10 runs. The results on the *vowel* database (Fig. 1(a)) are very interesting, since all the proposed strategies outperform the baseline case (which uses only the full and labelled samples in the initial reference set). Furthermore, the accumulated classification error decreases over time, which is a clear evidence of how the system is learning from the incoming, incomplete, unlabeled samples. Finally, in this case, where the delayed attribute was the one with the most relevance, the strategies *Put-and-reclassify* and *Wait-and-classify* can be seen to work better than *Do-nothing*. A likely explanation for this behavior is that, since the attribute is important for the correct classification, it is worth waiting for the delayed attribute to arrive either for reclassifying the object (*Put-and-reclassify*), or for incorporating the object into the reference set only once it is complete (*Wait-and-classify*).

Results for the *image* database (Fig. 1(b)) illustrate again how the considered strategies can boost the classification performance with respect to the conservative baseline approach. Interestingly, it is the *Do-nothing* strategy which now behaves better than the other two. Since the delayed attribute was the least relevant, it might happen that this attribute is hindering the classifier rather than helping it. As a consequence, and in the context of the *projection* technique that is being used in this work, it turns out to be better to passively ignore the attribute when it arrives than trying to make the most of it.

While in all examples above the missing attribute was delayed $\tau = 5$ time steps, it is interesting to evaluate how the actual delay affects the performance of the strategies under analysis. To this end, the same testing procedure was repeated for $\tau \in \{5, 15, 30, 45\}$ for several data sets. It was found that the *Do-nothing* and *Put-and-*

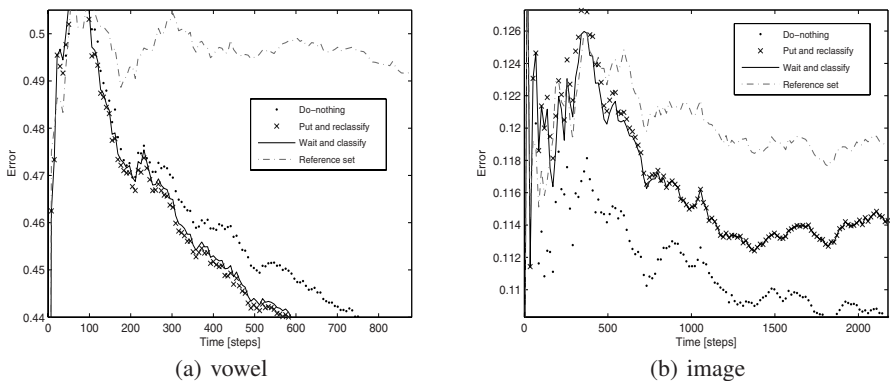


Fig. 1. Average error computed for ten runs. The baseline case (i.e., using only the initial reference set) is included for comparison with an off-line strategy. The missing attribute was either the most (a) or the least relevant (b).

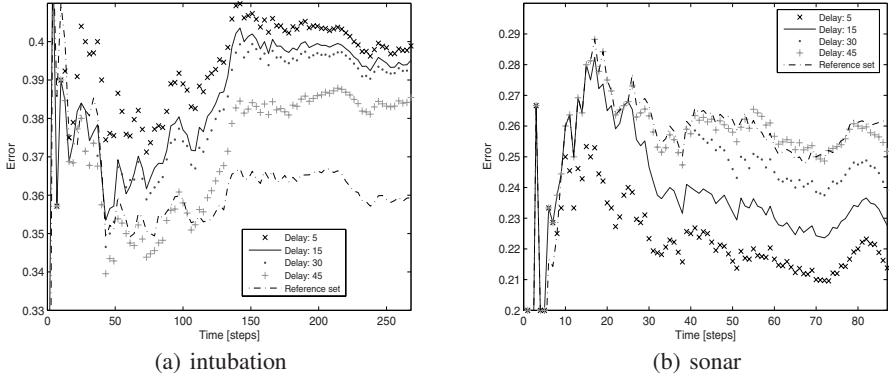


Fig. 2. Average accumulated classification errors, using the *Wait-and-classify* strategy, when the value of the missing attribute arrived $\tau = 5$, $\tau = 15$, $\tau = 30$ and $\tau = 45$ time steps after the incomplete object. In both figures the delayed attribute corresponds to the most relevant.

reclassify strategies did not exhibit a significant performance difference for distinct delays. However, differences were observed for the *Wait-and-classify* strategy, as illustrated in Fig. 2 for two of the tested databases.

In those data sets where this strategy did not work well, such as *intubation* (Fig. 2(a)), the accumulated error decreased when the delay increased. This can be explained as follows: the missing attribute happens to be unimportant (even harmful) and therefore, the longer it takes the attribute to arrive, the longer it takes to be incorporated into the object (and then into the training set) and thereby, less time it is affecting in the classification of subsequent objects. However, in cases such as *sonar* (Fig. 2(b)), where this strategy tends to work well, the more the delay, the higher the error. In this situation, the missing attribute appears to be necessary for the correct prediction of incoming objects.

4 Conclusions and Further Extensions

We have explored three strategies for the classification of streaming data with a single missing attribute. More specifically, we have presented a preliminary study for handling on-line data where the complete attribute vector arrives with a constant delay. Despite their simplicity, the results of the three strategies have shown some gains in performance when compared to the use of the initial reference set. Although these benefits are still marginal, the most important finding is that it seems possible to design some method to consistently handle the incomplete data in on-line classification of data streams.

The ultimate purpose of this work was to describe a novel, relevant problem that can be present in some real-world applications. Our study has revealed several interesting research directions regarding the classification of streaming (and incomplete) unlabeled data, such as: (i) An analysis of how the relevance of the missing attribute(s) affects the different classification strategies; (ii) The design of more elaborated methods for early classification of streaming data; (iii) The study of the benefits of different techniques

for handling missing attributes; (iv) An analysis of the case where the environment does change with time, and the reference sets will have to track these changes; and (v) An exploration of a more general situation with more than one delayed attribute and varying time delays.

Acknowledgment

This work has been supported in part by the Spanish Ministry of Education and Science under grants DPI2006–15542, CSD2007–00018 and TIN2009–14205.

References

1. Agarwal, C.: On-Demand Classification of Data Streams. In: Proc. ACM International Conference on Knowledge Discovery and Data Mining, pp. 503–508 (2004)
2. Agarwal, C.: Data Streams: Models and Algorithms. Springer, New York (2007)
3. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository, School of Information and Computer Science. University of California, Irvine, CA (2007), <http://archive.ics.uci.edu/ml/>
4. Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and Issues in Data Stream Systems. In: Proc. 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 1–16 (2002)
5. Bruzzone, L., Roli, R., Serpico, S.B.: An Extension of the Jeffreys–Matusita Distance to Multiclass Cases for Feature Selection. *IEEE Trans. on Geoscience and Remote Sensing* 33(6), 1318–1321 (1995)
6. Ganti, V., Gehrke, J., Ramakrishnan, R.: Demon: Mining and Monitoring Evolving Data. *IEEE Trans. on Knowledge and Data Engineering* 13(1), 50–63 (2001)
7. Gelman, A., Meng, X.L.: Applied Bayesian Modeling and Causal Inference from Incomplete Data Perspectives. John Wiley & Sons, Chichester (2004)
8. Kuncheva, L.I.: Classifier Ensembles for Detecting Concept Change in Streaming Data: Overview and Perspectives. In: Proc. 2nd Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications, pp. 5–10 (2008)
9. Maimon, O., Rokach, L.: Data Mining and Knowledge Discovery Handbook. Springer Science+Business Media, New York (2005)
10. Marwala, T.: Computational Intelligence for Missing Data Imputation, Estimation and Management: Knowledge Optimization Techniques. Information Science Reference (2009)
11. Muthukrishnan, S.: Data Streams: Algorithms and Applications. *Foundations and Trends in Theoretical Computer Science* 1(2), 117–236 (2005)
12. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge (1996)
13. Little, R.J.A., Rubin, D.B.: Statistical Analysis with Missing Data. Wiley, New York (1987)
14. Street, W.N., Kim, Y.: A Streaming Ensemble Algorithm (SEA) for Large-Scale Classification. In: Proc. 7th International Conference on Knowledge Discovery and Data Mining, pp. 377–382 (2001)
15. Takeuchi, J., Yamanishi, K.: A Unifying Framework for Detecting Outliers and Change Points from Time Series. *IEEE Trans. on Knowledge and Data Engineering* 18(4), 482–492 (2006)
16. Tsymbal, A.: The Problem of Concept Drift: Definitions and Related Work. Technical Report. Department of Computer Science, Trinity College, Dublin, Ireland (2004)
17. Vázquez, F., Sánchez, J.S., Pla, F.: A Stochastic Approach to Wilsons Editing Algorithm. In: Proc. 2nd Iberian Conference on Pattern Recognition and Image Analysis, pp. 35–42 (2005)

Exchange Rate Forecasting Using Classifier Ensemble

Zhi-Bin Wang¹, Hong-Wei Hao¹, Xu-Cheng Yin¹, Qian Liu¹, and Kaizhu Huang²

¹ Department of Computer Science, School of Information Engineering,
University of Science and Technology Beijing, Beijing 100083, China

² Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China
wzb1818@yahoo.cn, hhw@ies.ustb.edu.cn,
xuchengyin@ies.ustb.edu.cn, lq60001q@163.com,
kzhuang@nlpr.ia.ac.cn

Abstract. In this paper, we investigate the impact of the non-numerical information on exchange rate changes and that of ensemble multiple classifiers on forecasting exchange rate between U.S. dollar and Japanese yen. We first engage the fuzzy comprehensive evaluation model to quantify the non-numerical fundamental information. We then design a single classifier, addressing the impact of exchange rate changes associated with this information. In addition, we also propose other different classifiers in order to deal with the numerical information. Finally, we integrate all these classifiers using a support vector machine (SVM). Experimental results showed that our ensemble method has a higher degree of forecasting accuracy after adding the non-numerical information.

Keywords: exchange rate, forecasting, non-numerical information, support vector machine, classifier ensemble.

1 Introduction

With the development of economic globalization, exchange rate, as an important link of the international economic relations, is becoming more and more important. As a consequence, analyzing and forecasting the exchange rate accurately is of great significance, especially for making policies and investment decisions.

However, predicting exchange rate has always been a very difficult task. It involves a large number of economic, political, military and other factors. Currently, the neural network has been widely used to forecast exchange rate and become the main method for forecasting. In the literature, different structures of neural networks have been adopted and they can usually achieve remarkable results [1-5]. De Matos compared the performance of the multilayer feedforward network (MLFN) and the recurrent neural network (RNN) with the Japanese Yen Futures forecasting. Hsu and others developed a clustering neural network (CNN) to forecast the direction of movement of U.S. Dollar against German Mark. Similarly, combining the genetic algorithm and the neural networks, Shazly designed a hybrid neural network to predict the three-month forward rate of the Pound, German Mark, Japanese Yen and Swiss Franc.

From the study above, the predicting accuracy obtained with a certain neural network is usually higher than that of the traditional statistical prediction models and the

random walk model. However, there are still many limitations to this family of methods. First, these neural network based methods usually adopt a single classifier model, which may not be able to deal with a large number of input features adequately and appropriately. More importantly, due to the limited capacity of a single neural network and its inherit nature, these methods can merely utilize the quantified information. As a result, the non-numerical information, proven to be critical for the prediction accuracy, is usually discarded.

In order to deal with these problems, we adopt an integration method so as to combine multiple classifiers. We investigate the impact of non-numerical information for the exchange rate changes and engage a classifier ensemble method to forecast exchange rate between U.S. dollar and Japanese yen. First, we exploit the fuzzy comprehensive evaluation model to quantify the non-numerical fundamental information. Second, the corresponding single classifier is designed to present the impact of exchange rate changes with this information. In addition, other different classifiers are also proposed to deal with the numerical information. Finally, all these classifiers are integrated with a support vector machine (SVM).

The rest of the paper is organized as follows. Section 2 describes the non-numerical information quantification. And multiple classifiers ensemble is shown in Section 3. Section 4 describes experiments of forecasting exchange rate between U.S. dollar and Japanese yen. Finally, some conclusions and final remarks are set out in Section 5.

2 The Non-numerical Information Quantification

Exchange rate forecasting is a complex problem and involves many factors. Previous approaches merely adopt the numerical information. The non-numerical information which is difficult to be quantified is always discarded. However the impact of exchange rates changes with this information is also important and hence cannot be ignored.

2.1 The Non-numerical Information Selection

With analyzing the theories of exchange rate determination and considering the significant impacts on exchange rate changes, we select several important non-numerical information items which mainly include the following six aspects: government and banking policy, market psychology, news media, oil price, political situation and unexpected factors.

The above information is mainly from the following websites:

<http://edu.xinhuaonline.com>; <http://www.reuters.com/>; <http://www.fx185.com/>.

We collected a total of 85 trading days of the relevant information from January 1 to May 2, 2008 about the United States and Japan, and study the non-numerical information for the impact of exchange rate changes between the two currencies.

2.2 The Non-numerical Information Quantification

The non-numerical information is quantified by the fuzzy comprehensive evaluation model [6]. The process is described as follows.

First, calculate the weight of the non-numerical information with the binary comparison method. Then, evaluate the degree of membership according to the size of the affection to exchange rate changes. At last, quantify the non-numerical based on the weight and the degree of membership.

After calculation and analysis, the degree of impact is described as follows: Government and banking policy > market psychology > news media > oil price > political situation > unexpected factors. According to this relationship, the weight of the non-numerical information is calculated. The results are shown in Table 1.

Table 1. The matrix table and weight

Indicators	Government and banking policy	Market psychology	News media	Oil price	Political Situation	Unexpected factors	Σ	Weight
Government and banking policy	1	1	1	1	1	1	6	0.286
Market psychology	0	1	1	1	1	1	5	0.238
News media	0	0	1	1	1	1	4	0.190
Oil price	0	0	0	1	1	1	3	0.143
Political situation	0	0	0	0	1	1	2	0.095
Unexpected factors	0	0	0	0	0	1	1	0.048
Σ	1	2	3	4	5	6	21	1

While computing the degree of membership, we mainly consider the impact of exchange rate changes. If the information is favorable to the exchange rate rising, its degree of membership is close to 1. Contrarily, it is close to 0. If the change of exchange rate is not obvious, it is 0.5. Because the exchange rate involves two countries, the membership degree of the two countries is contrary in the table.

At last, we obtain the quantified value of this information with the weight and the degree of membership. Partial results are shown in Table 2.

3 Classifier Ensemble for Exchange Rate Forecasting

The exchange rate forecasting with multiple classifiers is proposed to deal with exchange rate changes with many economic indicators. First, we use different classifiers to deal with the economic indicator. Then we integrate these results from each single classifier. The classifier ensemble structure is shown in Fig. 1.

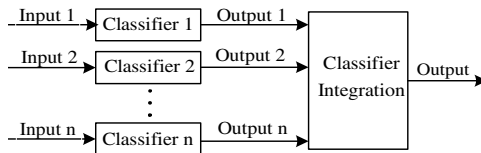


Fig. 1. The structure of the multiple classifiers system

Table 2. The non-numerical information quantification results

Information	America						Quantify value
	Government and banking policy	Market psychology	News media	Oil price	Political situation	Unexpected factors	
Weight	0.286	0.238	0.190	0.143	0.095	0.048	
1/2/2008	0.5	0.4	0.5	0.3	0.5	0.5	0.4719
1/3/2008	0.5	0.6	0.5	0.6	0.5	0.5	0.5224
1/4/2008	0.6	0.5	0.6	0.6	0.5	0.4	0.5243
1/7/2008	0.5	0.5	0.5	0.7	0.5	0.4	0.5152
1/8/2008	0.5	0.5	0.6	0.3	0.4	0.5	0.5081
1/9/2008	0.5	0.5	0.6	0.4	0.6	0.4	0.5180
1/10/2008	0.5	0.4	0.4	0.6	0.5	0.5	0.4886
1/11/2008	0.5	0.3	0.4	0.6	0.5	0.5	0.4691
1/14/2008	0.5	0.6	0.5	0.4	0.6	0.5	0.5176
1/15/2008	0.5	0.4	0.4	0.7	0.5	0.5	0.5100
1/16/2008	0.5	0.4	0.4	0.6	0.4	0.5	0.4605
1/17/2008	0.5	0.4	0.4	0.6	0.5	0.5	0.4786
1/18/2008	0.7	0.3	0.4	0.4	0.5	0.5	0.4806
1/22/2008	0.7	0.4	0.4	0.4	0.5	0.5	0.4701
1/23/2008	0.5	0.6	0.5	0.6	0.5	0.5	0.5095
1/24/2008	0.7	0.6	0.4	0.3	0.5	0.5	0.5262
1/25/2008	0.5	0.4	0.5	0.4	0.5	0.5	0.4848
...

3.1 Integrated Forecast

The integrated forecast is based on the intuitive idea that by combining several separate prediction models, the forecasting effect may be better than a single one [7].

In the experiments, denote there are n separate classifiers to predict, and for any input x , the output is $f_i(x)$, associated with the i classifier. The integrated prediction model is displayed as follows.

$$\tilde{f}(x) = \sum_{i=1}^n w_i f_i(x). \tag{1}$$

where, the $\tilde{f}_i(x)$ is the results of integrated forecasting, the weight of each individual classifier in integrated forecast is $w_i (i = 1, 2, \dots, n)$, $0 \leq w_i \leq 1$, $\sum_{i=1}^n w_i = 1$.

The method of integrated forecast has advantages in reducing the variance of the forecasting error. However, how to calculate the weight and which integrated approach to be adopted are difficult problems. At present, there are two categories of methods, the linear integration and the non-linear integration. We use the integration of support vector regression prediction model to forecast the exchange rate.

3.2 Ensemble Forecast with Support Vector Machine

In order to overcome the problems caused by the neural networks, the SVM based method is designed [8-9]. It uses the support vector regression (SVR) to solve the

problem of weight in Eq. (1). The basic idea is using the structural risk minimization to obtain the weight vector of the integrated forecast.

In fact, the support vector regression ensemble can be seen as a non-linear processing system [10]. It is displayed as following.

$$\tilde{y} = f(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n) \tag{2}$$

where, $(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)$ is the forecast result of separate prediction model, \tilde{y} is the integrated forecast result, $f(\bullet)$ is a non-linear function which is confirmed by the SVR. The solving steps are described as follows.

First, regress the forecast result of separate prediction model. Then, turn the results to the support vector using the kernel function. At last, study and output the optimal solution. The structure is shown in Fig. 2.

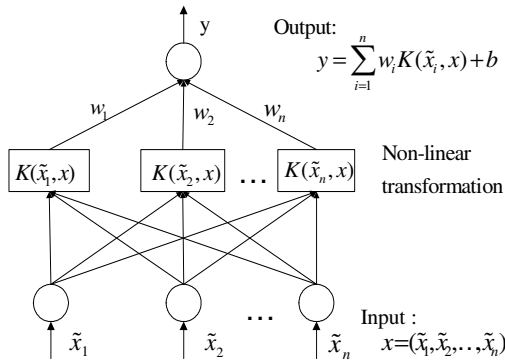


Fig. 2. The integration of support vector regression prediction model

4 Experiments and Analysis

4.1 Data Sources

In this paper, we select U.S. Dollar / Japanese Yen to forecast the exchange rate. The experimental data are from the website: <http://www.federalreserve.gov/release/>.

We use the data of 61 days from January 2 to March 31, 2008 except for Weekends and holidays as the training set to establish the multi-classifier system. Similarly, we adopt the 24 days from April 1 to May 2, 2008 as the testing set. Specific data can be retrieved from the related websites.

4.2 Data Preprocessing and Evaluation Criteria

We first normalize the raw data of the exchange rate and transfer them into the special form which is suitable for the neural network processing. The data are mapped to [0.1, 0.9] according to repeatedly tested and compared in the experiments. The mapping function is described as follows.

$$y = \frac{(x - \min)(h - l)}{\max - \min} + l. \tag{3}$$

where, y is the standardized data, x denotes the raw data, \max is the largest data of the raw data, \min is the smallest one and h is the upper bound of a specific interval as well as l is the lower bound, $0.1 \leq l \leq h \leq 0.9$.

In order to evaluate the forecast performance, we use the Mean Absolute Error (MAE) and the Direction Accuracy (DA) as the evaluation criteria. The formula for calculating MAE is as follows.

$$MAE = \frac{1}{N} \sum_{i=1}^N |x_i - \tilde{x}_i|. \tag{4}$$

where, \tilde{x}_i is the forecast result, x_i is the actual value, N is the forecast period.

However, from the perspective of investors, the MAE can not bring direct suggestions to their investment. Due to the direction of exchange rate changes is more important for them to make decisions. So, we use the DA to evaluate the effect of forecasting.

$$DA = \sum_{i=1}^N \frac{A_i}{N}. \tag{5}$$

Here, if $(x_{i+1} - x_i)(\tilde{x}_{i+1} - x_i) \geq 0$, $A_i = 1$. Else, $A_i = 0$.

4.3 Experimental Results and Analysis

4.3.1 The Effectiveness of Non-numerical Information

The quantified non-numerical information (in Table. 2) can be directly integrated with the exchange rate value and numerical index. In order to verify the effective of the quantitative methods and the selected information, we use the SVM and the RBF (Radial Basis Function) to carry out the experiment. Results are shown in Table 3 and Table 4 respectively.

Table 3. The forecast results of SVM

Input variables	Not the non-numerical information		The non-numerical information	
	MAE	DA	MAE	DA
Exchange value	0.0574	52.17%	0.0467	66.96%
Exchange value and the large deposit rates	0.0462	53.12%	0.0369	86.96%
Exchange value and the bond yields	0.0480	43.48%	0.0370	78.27%
Exchange value and the lending rate	0.0472	50.05%	0.0373	70.83%
Exchange value and the treasury rates	0.0509	53.41%	0.0403	65.22%
Exchange value and the federal funds rate	0.0465	50.15%	0.0372	75.12%

Table 4. The forecast results of RBF

Input variables	Not the non-numerical information		The non-numerical information	
	MAE	DA	MAE	DA
Exchange value	0.0674	52.17%	0.0572	66.96%
Exchange value and the large deposit rates	0.0467	52.36%	0.0368	82.61%
Exchange value and the bond yields	0.0543	47.83%	0.0405	73.91%
Exchange value and the lending rate	0.0431	51.18%	0.0432	65.22%
Exchange value and the treasury rates	0.0508	53.62%	0.0382	83.58%
Exchange value and the federal funds rate	0.0438	54.17%	0.0384	66.67%

Observed from the tables, the effect of forecast is better if we add the economic indicators. Moreover, with the non-numerical information, not only the MAE indicator is significantly lower, but also the DA is improved greatly. It fully demonstrates that the six types of information, as well as quantitative methods are effective.

4.3.2 The Integrated Forecast Results

We use the above method to forecast the exchange rate. The output of single classifiers forms a feature vector and the integration is seen as a secondary forecast.

In the experiment, we adopt one single classifier to process every different economic indicator. The process is showed as follows: (1) study the implicit principles with single classifier, (2) integrate these forecast results from every single classifier, and (3) finally obtain the forecast results. The experimental results are shown in Table 5.

Table 5. The integrated forecast results

Input variables	Not the non-numerical information		The non-numerical information	
	MAE	DA	MAE	DA
Integrate all the indicators	0.0393	78.26%	0.0367	86.96%

If we compare Table 5 with Table 3 and Table 4, the forecasting accuracy in terms of both MAE and DA is observed to be greatly improved and is much better than that of the single classifier. Additionally, the non-numerical information can also benefit the exchange rate forecasting. The results show that the integrated network can utilize different type of neural network architectures and information so as to make a better forecasting. It can overcome the defects of the single classifier and consequently achieves better performance.

5 Conclusion

In this paper, we adopt the integration method of multiple classifiers to forecast the exchange rate. We have investigated the impact of the non-numerical information on

the exchange rate changes between the dollar and Japanese yen. Experimental results showed that our integrated method effectively improved the performance of exchange rate forecast. Specially, the DA is improved greatly. However, how many economic indicators to be integrated for reaching the best effect of forecasting is still an open problem. We are aware of that, with the global financial crisis intensifying, the exchange rate forecasting is a big challenge, which needs more theories, methods, and technologies from all related fields, such as economics, mathematics, and computer science. We will investigate these issues in the future.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No.60675006 and thanks Liang Jia for his contributions to this paper.

References

1. Tenti, P.: Forecasting foreign exchange rates using recurrent neural networks. *Applied Artificial Intelligence* 10, 567–581 (1996)
2. El Shazly, M.R., El Shazly, H.E.: Forecasting currency prices using a genetically evolved neural network architecture. *International Review of Financial Analysis* 8, 62–82 (1999)
3. Leung, M.T., Chen, A.-S., Daouk, H.: Forecasting exchange rates using general regression neural networks. *Computers & Operations Research* 27, 1093–1109 (2000)
4. Reddy, M.K., Kumar, B.N.: Forecasting foreign exchange rates using time delay neural networks. In: *The 2008 International Conference on Data Mining*, pp. 181–194. IEEE Press, Las Vegas (2008)
5. Kasuga, H.: Exchange rates and ownership structure of Japanese multinational firms. *Japan and the World Economy* 10, 661–678 (2008)
6. Xu, Y.-T., Wang, L.-S.: Fuzzy comprehensive evaluation model based on rough set theory. In: *5th International Conference on Cognitive Informatics*, pp. 877–880. IEEE Press, Beijing (2006)
7. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. In: *Advances in Neural Information Processing Systems*, pp. 231–238. MIT Press, Cambridge (1995)
8. Kamruzzaman, J., Sarker, R.A., Ahmad, I.: SVM based models for predicting foreign currency exchange rates. In: *Third International Conference on Data Mining*, pp. 557–560. IEEE Press, Los Alamitos (2003)
9. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, 2nd edn. John Wiley & Sons, San Francisco (2001)
10. Wang, S.-Y., Yu, L.-A., Lai, K.-K.: TEI@I Methodology and Its Application to Exchange Rates Prediction. *Chinese Journal of Management* 4, 21–27 (2007)

Erratum to “Backpropagation Learning Algorithm for Multilayer Phasor Neural Networks” by Gouhei Tanaka and Kazuyuki Aihara

The part from the last equation in page 490 should be corrected as follows:

—— **incorrect** ——

$$i_l = \sum_{k=1}^{N_k} l_k i_k = \sum_{k=1}^{N_k} l_k \cos \theta_k + \sum_{k=1}^{N_k} l_k \sin \theta_k$$

which leads to

$$\tan \theta_l = \frac{l \sin \theta_l}{l \cos \theta_l} = \frac{\sum_k l_k \sin \theta_k}{\sum_k l_k \cos \theta_k}$$

By differentiating both sides with respect to θ_k , we obtain

$$\frac{1}{\cos^2 \theta_l} \frac{d\theta_l}{d\theta_k} = \frac{(-l_k \cos \theta_k)(l \cos \theta_l) + (l \sin \theta_l)(l_k \sin \theta_k)}{l^2 \cos^2 \theta_l}$$

Hence,

$$\frac{d\theta_l}{d\theta_k} = \frac{l_k \sin \theta_k \sin \theta_l + l_k \cos \theta_k \cos \theta_l}{l}$$

—— **correct** ——

$$i_l = \sum_{k=1}^{N_k} l_k i_k = \sum_{k=1}^{N_k} l_k \cos \theta_k + \sum_{k=1}^{N_k} l_k \sin \theta_k$$

which leads to

$$\tan \theta_l = \frac{l \sin \theta_l}{l \cos \theta_l} = \frac{\sum_k l_k \cos \theta_k + \sum_k l_k \sin \theta_k}{\sum_k l_k \cos \theta_k}$$

By differentiating both sides with respect to θ_k , we obtain

$$\frac{1}{\cos^2 \theta_l} \frac{d\theta_l}{d\theta_k} = \frac{(l_k \sin \theta_k + l_k \cos \theta_k)(l \cos \theta_l) + (l \sin \theta_l)(l_k \sin \theta_k + l_k \cos \theta_k)}{l^2 \cos^2 \theta_l}$$

Hence,

$$\frac{d\theta_l}{d\theta_k} = \frac{l_k \cos(\theta_l - \theta_k) + l_k \sin(\theta_l + \theta_k)}{l}$$

Author Index

- Abdull Hamed, Haza Nuzly II-611
Aguilera, Emanuel II-11
Ahamdi, Majid I-701
Aihara, Kazuyuki I-143, I-484, II-401, E1
Alecú, Lucian I-135
Alnajjar, Fady I-451, II-65
Aly, Saleh I-733
Amari, Shun-ichi I-185, I-649
Amemiya, Yoshihito II-384
Amornsamankul, Somkid I-554
An, Dong Un II-281
Ando, Ruo II-540
Andrews, Emad A.M. I-100
Aoki, Kenji I-19
Araujo, Allan David Garcia I-229
Arik, Sabri I-460
Asai, Tetsuya II-384
Asirvadam, Vijanth S. I-126
Assawamakin, Anunchai II-493
- Bae, Yong Chul I-759
Ban, Sang-Woo I-693
Ban, Tao II-520, II-530, II-729
Bando, Takashi I-468
Barakova, Emilia II-430
Barczak, Andre L.C. II-675
Bedingfield, Susan II-770
Bellas, Francisco II-75
Ben-Tal, Gadi II-675
Bennani, Younès I-546
Bertoni, Fabiana Cristina I-267
Besse, Camille I-237, I-433, II-648
Billings, Stephen I-34, I-57
Binwahlan, Mohammed Salem II-216
Boné, Romuald II-786
Bonner, Anthony J. I-100
Borland, Ron II-770
Bui, Michael II-812
- Cabanes, Guénaël I-546
Cai, Youbo II-420
Caiafa, Cesar F. I-221
Cardot, Hubert II-786
Castellano, Marcello II-777
Cerrato, Mario I-441
- Cha, En-Jong I-630
Chaib-draa, Brahim I-237, I-433, II-648
Chaiyaratana, Nachol II-493
Chang, Yongping I-340
Char, ByungRea II-281
Chen, Cunbao II-746
Chen, Guici I-259
Chen, Junfei II-762
Chen, Ping I-247
Chen, Qiaona II-289
Chen, Ye II-520
Cherif, Aymen II-786
Cheung, Wai Keung II-273
Cho, Minkook I-716
Cho, Sung-Bae II-630
Choe, Yoonsuck I-302
Choge, H. Kipsang I-520, I-639
Choi, Choong Hwan I-365
Choi, Heeyoul I-302
Choi, Sangbok I-759
Choi, Seungjin I-175, I-302
Chongstitvatana, Prabhas II-122
Chow, Tommy W.S. I-212
Chumwatana, Todsanai II-691
Cichocki, Andrzej I-221, I-323,
I-409, I-538
Cihan, Ahmet I-846
Cobos, Maximo II-11
Coca, Daniel I-34, I-57
Coelho, André L.V. I-512
Coghill, Ken II-770
Creighton, Doug I-285, II-141
Cuzzola, Maria II-360
Cyganek, Boguslaw I-399
- Dai, Xinyu II-754
Dallaire, Patrick I-433
Dehzangi, Abdollah II-503
Demiriz, Ayhan I-846
Deng, Aidong II-738
de Peretti, Christian I-441
Ding, Xiaojiang II-770
Do, Hai Thanh II-465
Domínguez, Enrique I-743

- Doria Neto, Adriaio Duarte I-229
 Doya, Kenji II-638
 Dozono, Hiroshi II-836
 Duch, Włodzisław II-206
 Duro, Richard J. II-75

 Ebrahimpour, Reza II-439
 Egerton, Simon II-93
 Elfwing, Stefan II-638
 Eto, Masashi II-556, II-565

 Farhoudi, Zeinab II-439
 Feng, Zuren II-253
 Fiasché, Maurizio II-360
 Fong, Anthony S.S. I-212
 Fooprateepsiri, Rerkchai I-788
 Frezza-Buet, Hervé I-135
 Fricout, Gabriel I-502
 Friederich, Uwe I-34
 Fujimura, Kikuo II-794
 Fujiwara, Kantaro I-143
 Fukaya, Naoki I-468
 Fukuda, Wataru II-19
 Fukumi, Minoru I-520, I-639
 Funase, Arao I-409
 Fung, Chun Che I-554
 Furber, Steve I-425

 Gajate, Agustin II-573
 Galluppi, Francesco I-425
 Ganapathy, Velappa II-93
 Gao, Ji I-389
 Gao, Ya II-226
 Geist, Matthieu I-502
 Glette, Kyrre II-159
 Gonda, Eikou II-794
 Goto, Yoshinobu II-299
 Guo, Ping I-373, I-778
 Guo, Shanqing II-729
 Guo, Xinjian I-798
 Guo, Zhenya I-151

 Haber, Rodolfo II-573
 Habu, Manabu II-449
 Hafiz, Abdul Rahman I-451, II-65
 Hao, Hong-Wei I-838, I-884
 Harada, Naoyuki I-167
 Hardie, Roger C. I-57
 Haruechaiyasak, Choochart II-309
 Hasegawa, Osamu I-769
 Hassab Elgawi, Osman II-83

 Hatagami, Yutaro II-352
 Hayashi, Hirotomo I-657
 Hayashi, Satoshi II-556
 Hazeyama, Hiroaki II-548
 Hidaka, Akinori II-38
 Hirata, Yutaka I-84
 Hirose, Akira II-263
 Hishiki, Tetsuya II-392
 Hitomi, Kentaro I-468
 Ho, Kevin I-494
 Holeña, Martin II-131
 Horio, Keiichi II-449
 Horiuchi, Tadashi II-874
 Hosaka, Ryosuke II-401
 Høvin, Mats II-159
 Hu, Yingjie II-483
 Hu, Zhiwei I-819
 Huang, Chung-Hsien II-512
 Huang, Jiangshuai I-10
 Huang, Kaizhu I-838, I-884
 Huang, Mao Lin II-699
 Huang, Tingwen I-194
 Husselmann, Alwyn II-667

 Iacopino, Pasquale II-360
 Ichimaru, Kota I-622
 Iima, Hitoshi II-169
 Ikeda, Kazushi I-468
 Inagaki, Keiichiro I-84
 Inoue, Daisuke II-556, II-565
 Inoue, Hirotaka II-820
 Intan, Rolly II-720
 Ishihara, Akito I-84
 Ishii, Kazuo II-409
 Ishii, Shin I-590, I-598, II-19
 Ishikawa, Masumi II-420
 Ishikawa, Yuta I-159, I-167
 Islam, Tanvir I-26
 Ito, Takaichi II-828
 Ito, Yoshifusa I-417
 Itoh, Hideaki I-19
 Itoh, Yoshio II-874
 Izumi, Hiroyuki I-417

 Jaffry, S. Waqar I-72
 Janjarasjitt, Suparek II-326
 Jaruszewicz, Marcin II-601
 Jeong, Dongmin I-381
 Ji, Guoli I-151
 Ji, Zheng I-685
 Jiao, Weidong I-340

- Jin, Xin I-425
 Jin'no, Kenya II-234
 Jinarat, Supakpong II-309
 Juusola, Mikko I-34, I-57

 Kabir, Md. Monirul II-150, II-242
 Kadobayashi, Youki II-520, II-530,
 II-548, II-729
 Kaji, Daisuke I-476
 Kakihara, Toshiyuki II-794
 Kamiji, Nilton L. I-84, II-189
 Kamiyama, Yoshimi I-84
 Kanemura, Atsunori II-19
 Kang, Yoonseop I-175, I-302
 Kannon, Takayuki I-84
 Karungaru, Stephen I-520, I-639
 Kasabov, Nikola II-114, II-360, II-483,
 II-520, II-530, II-611
 Katake, Anup I-302
 Kato, Satoru II-874
 Kawewong, Aram I-769
 Khan, Mukaram I-425
 Kho, Henry II-114
 Khosravi, Abbas I-285, II-141
 Kijisrikul, Boonserm I-708, II-583
 Kikombo, Andrew Kilinga II-384
 Kim, Bumhwi I-693
 Kim, Chul-Won II-281
 Kim, Dami I-649
 Kim, Kyung-Joong II-630
 Kim, Soohyung I-810
 Kim, Ungmo I-312
 Kim, Younghee I-312
 King, Irwin I-866
 Kinoshita, Kai II-367
 Kitahara, Kunio II-263
 Kobayashi, Kunikazu I-530
 Kohata, Yasushi I-606
 Kondo, Toshiyuki II-179
 Koshiyama, Yohei II-56
 Kraipeerapun, Pawalai I-554
 Kula, Ufuk I-846
 Kurata, Masahumi II-794
 Kuremoto, Takashi I-530
 Kurihara, Masahito I-606
 Kurita, Takio II-38
 Kuroe, Yasuaki I-667, II-169
 Kurogi, Shuichi I-622, II-56
 Kurosu, Chihiro II-234
 Kurutach, Werasak I-788

 Lee, Guesang I-810
 Lee, Jiann-Der II-512
 Lee, Kyung-Hee I-630
 Lee, Minhø I-381, I-693, I-759, II-1
 Lee, S.T. II-512
 Lee, Sang-Kwang I-829
 Lee, Soo-Young I-365, I-649
 Lee, Yonggon I-381
 Leung, Chi Sing I-277, II-273
 Li, Pengfei II-107
 Li, Zhiqing I-751
 Li, Zhixin I-751
 Liao, Weihao II-762
 Lim, Chee Peng II-475, II-593
 Lim, Young-Chul II-1
 Limwongse, Chanin II-493
 Linke, David II-131
 Liu, Guoqing I-357
 Liu, MingHui II-675
 Liu, Qian I-838, I-884
 Liu, Yi II-845
 Loparo, Kenneth A. II-326
 Lopez, Jose J. II-11
 Lourens, Tino II-430
 Lu, Bao-Liang I-685
 Lu, Hongtao I-819
 Lu, Jie II-226, II-318
 Lu, Ning II-318
 Lukas II-114
 Luque, Rafael Marcos I-743

 Maeda, Shin-ichi I-598, II-19
 Mahdian, Babak II-683
 Makaremi, Iman I-701
 Mańdziuk, Jacek II-601
 Maniwa, Yoshio II-794
 Manoonpong, Poramate II-47
 Marier, Jean-Samuel II-648
 Martins, Allan de Medeiros I-229
 Mashinchi, M. II-336
 Mashinchi, M.H. II-336
 Mastronardi, Giuseppe II-777
 Maszczyk, Tomasz II-206
 Matsuda, Nobuo II-802
 Matsuda, Yoshitatsu I-204
 Matsuka, Toshihiko II-352
 Matsumoto, Shuhei II-449
 Melkumyan, Arman I-331
 Michlovský, Zbynek II-530, II-611

- Millán-Giraldo, Mónica I-875
 Mima, Hiroki I-468
 Mirikitani, Derrick T. I-91
 Mirmotahari, Omid II-159
 Mitsudo, Takako II-299
 Mitsukura, Yasue I-520
 Miyamoto, Daisuke II-548
 Mizoue, Hiroyuki I-530
 Mohandesi, Ehsan II-503
 Monteiro, Fernando C. II-657
 Morabito, Francesco C. II-360
 Mori, Takeshi I-590
 Mori, Yoshihiro I-667
 Morita, Satoru II-28
 Mouri, Motoaki I-409
 Muñoz, José I-743
 Murase, Kazuyuki I-451, II-65, II-150,
 II-242
- Nagamatu, Masahiro II-457
 Nahavandi, Saeid I-285, II-141
 Naito, Takuto I-110
 Nakajima, Yoshitaka II-299
 Nakakuni, Masanori II-836
 Nakamura, Kiyohiko I-19
 Nakano, Ryohei I-159, I-167
 Nakao, Koji II-556, II-565
 Nakkrasae, Sathit I-554
 Nascimento, Diego S.C. I-512
 Navarro, Jose C. II-475
 Nettleton, Eric I-331
 Ng, Keng Hoong II-503
 Nguyen, Quang Vinh II-699
 Nguyen, Toan I-810
 Niu, Zhendong II-344
 Nozawa, Takayuki II-179
- Obayashi, Masanao I-530
 Ohkita, Masaaki II-794
 Ohtani, Taishi II-449
 Okamoto, Keisuke I-562
 Okumura, Manabu II-583
 Onoda, Tetsuya II-828, II-855
 Orgun, Mehmet A. II-336, II-699
 Quarbya, Lahcen I-91
 Oyabu, Matashige II-802
 Oyama, Tadahiro I-520, I-639
 Ozawa, Seiichi I-562
- Paik, Sang Kyoo I-759
 Palomo, Esteban José I-743
 Pan, Hongxia II-620
 Panda, Mrutyunjaya I-614
 Pang, Shaoning II-520, II-530
 Park, Hyeyoung I-716
 Park, Hyukro I-810
 Park, Jonghyun I-810
 Park, Sun II-281
 Patra, Manas Ranjan I-614
 Pears, Russel II-114
 Petrovic-Lazarevic, Sonja II-770
 Phan, Anh Huy I-221, I-323, I-538
 Phienthrakul, Tanasanee II-583
 Phon-Amnuaisuk, Somnuk I-570, I-580,
 II-503
 Phongpensri (Chantrapornchai),
 Chantana I-856
 Phongsuphap, Sukanya I-676
 Phoophuangpairoj, Rong I-676
 Pietquin, Olivier I-502
 Piroonratana, Theera II-493
 Postma, Marten I-57
 Prieto, Abraham II-75
 Pulikanti, Srikanth II-196
- Qiao, Deng-yu I-277
 Qiao, Xiaofei I-247
- Rast, Alexander I-425
 Remijn, Gerard B. II-299
 Reyes, Napoleon H. II-667, II-675
 Ripon, Kazi Shah Nawaz II-159
 Rungsawang, Arnon II-309
 Ryu, Joonsuk I-312
- Saic, Stanislav II-683
 Saito, Toshimichi I-118, II-234, II-376
 Sakai, Yutaka II-401
 Salim, Naomie II-216
 Sánchez, J. Salvador I-875
 Sangngam, Cholticha I-856
 Sato, Naoyuki I-49
 Sato, Seitaro I-622
 Satoh, Shunji I-84
 Scher, Mark S. II-326
 Sevgen, Selcuk I-460
 Shahjahan, Md. II-150, II-242
 Shamsuddin, Siti Mariyam II-611
 Shang, Lin II-754

- Shen, Yi I-259
Shi, Daming I-357
Shi, Zhichen I-798
Shi, Zhiping I-751
Shi, Zhongzhi I-751
Shibata, Tomohiro I-468
Shigang, Li II-794
Shin, Heesang II-667
Shouno, Hayaru I-84
Siani, Carole I-441
Silva, Ivan Nunes I-267
Singh, Alok II-196
Sinsomros, Saravudh II-493
Son, Kweon I-381
Song, Jungsuk II-556
Song, Qingsong II-253
Song, Shengli I-247
Song, Zhuoyi I-57
Sonoda, Kotaro II-565
Srinivasan, Cidambi I-417
Steinfeldt, Norbert II-131
Su, Shau-Chiuan II-512
Suanmali, Ladda II-216
Suh, Jae-Won I-630, I-829
Suh, Young-Ho I-829
Sum, John I-494
Sum, Pui Fai I-277
Sun, Shiliang I-349, II-289
Suzuki, Hideyuki I-143
Suzuki, Mio II-556
Swiecicki, Mariusz II-710
- Takahashi, Hiroki I-19
Takamura, Hiroya II-583
Takatsuka, Masahiro II-812, II-845
Takeichi, Hiroshige II-299
Takemura, Yasunori II-409
Takeuchi, Ichiro I-159, I-167
Takiguchi, Masao I-118
Takumi, Ichi I-409
Tamsumpaolerd, Sutthipong I-788
Tan, Choon Ling II-93
Tan, Kay Sin II-475
Tan, Shing Chiang II-475, II-593
Tanaka, Gouhei I-484, E1
Tang, Hesheng II-107
Tangwongsan, Supachai I-676
Taniguchi, Rin-ichiro I-733
Taniguchi, Tatsuki II-189
Tarricone, Gianfranco II-777
- Terada, Akira II-865
Teshima, Shimon II-376
Tjondronegoro, Dian I-724
Tobimatsu, Shozo II-299
Tokutaka, Heizo II-794, II-802
Tominaga, Kazuhiro II-449
Tomizawa, Hiroki I-598
Tongprasit, Nopparit I-769
Torikai, Hiroyuki II-367, II-392
Tørresen, Jim II-159
Traver, V. Javier I-875
Treur, Jan I-72
Tsang, Peter Wai-Ming II-273
Tsuge, Satoru I-520, I-639
Tsuruta, Naoyuki I-733
- Uchibe, Eiji II-638
Ueda, Naonori II-189
Usui, Shiro I-84, II-189
- Vega, Pastora II-573
Verma, Anju II-360
- Wakuya, Hiroshi II-865
Wang, Bin I-1
Wang, Dianhui II-465
Wang, Haipeng II-754
Wang, J.J. II-512
Wang, Yonggang II-253
Wang, Yongji I-10
Wang, Zhaoliang II-107
Wang, Zhi-Bin I-838, I-884
Watanabe, Sumio I-476
Wee, Hui-Ming II-226
Weerasinghe, Jagath II-457
Wei, Xin II-738
Wei, Xiuye II-620
Wei, You-You II-512
Weng, Yuan II-762
Widiputra, Harya II-114
Wiriathamabhum, Peratham I-708
Wong, Kok Wai II-691
Wongseeree, Waranyu II-493
Woo, Jeong-Woo II-1
Worasucheep, Chukiat II-122
Wörgötter, Florentin II-47
- Xia, Huaiying I-373
Xie, Hong II-691
Xu, Lu I-212
Xu, Qiuliang II-729

- Xu, Wen-Chuin II-512
 Xu, Xiaomei II-344
 Xu, Xin II-620
 Xu, Yang I-212

 Yamaguchi, Kazunori I-204
 Yamaguchi, Yoko I-26
 Yamakawa, Takeshi II-449
 Yamamoto, Masashi II-794
 Yamauchi, Koichiro I-606
 Yamazaki, Keisuke I-110
 Yang, Dong I-778
 Yang, Haiqin I-866
 Yang, Shixi I-340
 Yang, Zijiang I-151
 Ye, Congting I-151
 Yeh, Chung-Hsing II-770
 Yin, Cunyan II-754
 Yin, Jianping I-293
 Yin, Xu-Cheng I-838, I-884
 Yin, Yilong I-798
 Yoo, Hyang-Mi I-829
 Yoo, Jae-Kwon I-365
 Yoshioka, Katsunari II-565

 Young, David II-770
 Yu, Gang I-819
 Yu, Ying I-1
 Yucel, Eylem I-460
 Yuliana, Oviliani Yenty II-720
 Yuno, Hiroshi II-56

 Zeng, Zhi-Qiang I-389
 Zeng, Zhigang I-194
 Zhan, Yubin I-293
 Zhang, Guangquan II-226, II-318
 Zhang, Ke-Bing II-699
 Zhang, Ligang I-724
 Zhang, Liming I-1
 Zhang, Liqing I-221
 Zhang, Qingjiu I-349
 Zhao, Li II-738, II-746
 Zhao, Qiangfu I-657
 Zhao, Qibin I-221
 Zhao, Shihao II-762
 Zhou, Hanying I-10
 Zhou, Suiping I-357
 Zhu, Song I-259
 Zin, Indra Bin Mohd. II-65